DELL Technologies

White Paper

iDRAC9 User Interface to Redfish Mapping

This paper provides an overview of the iDRAC9 User Interface (UI) and how to extract same content using the Redfish Application Programing Interface (API).

July 2023, Version 1.3

Revisions

Date/Version	Description
Nov 2022/1.0	Initial release
Jan 2023/1.1	Updated sections: Cooling/Cooling Overview/Fan Status(Overall Fan Health in Note section) iDRAC Details/RAC Information/Hardware Version Alert Configuration/SNMP Traps Configuration/Test SNMP Trap Alert Configuration/SMTP(email) Configuration/Send Test email Alert Configuration/Alert Recurrence Cooling Configuration/Get Thresholds Minimum Fan Speed in PWM (% of Max) Cooling Configuration/Set Thresholds Minimum Fan Speed in PWM (% of Max) Management USB Settings/Get Device Present Cooling/Cooling Overview/PCIe Airflow Settings
Jan 2023/1.2	Added Cooling/Cooling Overview/Fan Status/Fan Type section
July 2023/1.3	Updated sections: Cooling/Cooling Overview/Fan Status/Fan Type Cooling/Cooling Overview/Average Fan Speed Memory/Memory Attributes/Installed Capacity Memory/Memory Attributes/Maximum Capacity Removable Media/Internal SD Module Status Power/Cumulative Reading/Time Power/Cumulative Reading/Total Usage Power/Historical Peaks/Time Power/Historical Peaks/Peak Watts

	Power/Historical Peaks/Peak Watts Time Added sections: Local Users/Get SSH Key Local Users/Upload SSH Key Local Users/Delete SSH Key Cooling/Cooling Overview/Redundancy Status Cooling/Cooling Overview/Net System Airflow RSA SecurID Configuration/RSA Server Certificate/Upload RSA Server Certificate RSA SecurID Configuration/RSA Server Certificate/Test Network Connection SSL/TLS Certificate Signing Request/CA Certificate/Upload CA Certificate Removable Media/Internal SD Module(Redundancy Status) Upload Custom Defaults Server Configuration Profile Locally Local Users/Get Smart Card User Certificate Local Users/Upload Smart Card User Certificate Local Users/Get Smart Card Trusted CA Certificate Local Users/Upload Smart Card Trusted CA Certificate Enclosure/Fans Enclosure/Power Supplies Enclosure/Temperature Probes Enclosure/EMM Get Automatic Schedule Update Settings Enable/Disable Automatic Update Set Automatic Schedule Update Clear Automatic Update Settings

Acknowledgments

Author: Texas Roemer, Rich Schnur, Neeraja Kothala

Support: Ajay Shenoy

Contents

Executive Summary

Dell PowerEdge servers offer a comprehensive range of embedded systems management functions with the Integrated Dell Remote Access Controller (iDRAC). These functions are designed by adhering to industry standard application programming interfaces (APIs) including Redfish. iDRAC technology is part of a large solution that helps keep business critical applications and workloads available always. The technology allows administrators to deploy, monitor, manage, configure, update, troubleshoot, and remediate Dell servers from any location, and without the use of agents. It accomplishes these tasks regardless of an operating system, hypervisor, presence, or state.

The Redfish Scalable Platforms Management API is an Open Standard the Distributed Management Task Force (DMTF) defines. Redfish is modern systems management interface standard. The Redfish API enables scalable, secure, and open server management. The Redfish API uses RESTful interface semantics to access data that is defined in model format to perform out-of-band systems management. It is for a wide range of servers ranging from stand-alone servers to rack mount and bladed environments and for large-scale and multicloud environments.

This document shows how the iDRAC9 Redfish API can be used to get data that is available from the iDRAC9 User Interface (UI).

Although the Redfish data model is schema-based, it is not necessary for new users to understand schema to use Redfish.

1 Section 1: iDRAC UI Login Page

1. Get Security Policy Message

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/GUI.1.SecurityPolicyMessage

Header: content-type application/json

Auth: Basic or X auth
```

2. Set Security Policy Message

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"GUI.1.SecurityPolicyMessage": "<pass in possible string
value>"}}
```

2 Section 2: Dashboard Page

1. Health Information System

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=Status

Header: content-type application/json

Auth: Basic or X token
```

2. Health Information Storage

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/StorageRollupSt
atus

Header: content-type application/json

Auth: Basic or X token
```

3. System information current server power state

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=PowerState

Header: content-type application/json

Auth: Basic or X auth
```

4. System information server model

```
Command: GET
```

```
URI: redfish/v1/Systems/System.Embedded.1?$select=Model

Header: content-type application/json

Auth: Basic or X auth
```

5. System information hostname

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=HostName

Header: content-type application/json

Auth: Basic or X auth
```

6. System information operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerOS.1.OSName

Header: content-type application/json

Auth: Basic or X auth
```

7. System information operating system version

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerOS.1.OSVersion

Header: content-type application/json

Auth: Basic or X auth
```

8. System information service tag

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Bios?$select=Attributes/SystemServiceTag

Header: content-type application/json

Auth: Basic or X auth
```

9. System information BIOS version

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=BiosVersion

Header: content-type application/json

Auth: Basic or X auth
```

10. System information iDRAC version

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1?$select=FirmwareVersion

Header: content-type application/json

Auth: Basic or X auth
```

11. System information IP address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.Address

Header: content-type application/json
```

```
Auth: Basic or X auth
```

12. System information MAC address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentNIC.1.MACAddress

Header: content-type application/json

Auth: Basic or X auth
```

13. System information license

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellLicenses

Header: content-type application/json

Auth: Basic or X auth
```

Note: You must parse the Members collection data for specific license details.

14. Task summary/Job Queue Details

```
Command: GET

URI: redfish/v1/JobService/Jobs?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

15. Recent logs/SEL logs

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/LogServices/Sel/Entries

Header: content-type application/json

Auth: Basic or X auth
```

16. Power operations

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "<allowable value>"}
```

Note: For supported allowable values, run GET on URI "redfish/v1/Systems/System.Embedded.1?$select=Actions" and view property "ResetType@Redfish.AllowableValues".

17. LED operations

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1

Header: content-type application/json

Auth: Basic or X auth

Body: {"IndicatorLED": "<allowable values: Lit or Blinking>"}
```

Note: iDRAC does not support DMTF Off value.

18. Reboot iDRAC

```
Command: POST
```

```
URI: redfish/v1/Managers/iDRAC.Embedded.1/Actions/Manager.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType":"GracefulRestart"}
```

19. Enable/Disable System Lockdown Mode

```
Command: PATCH

URI: redfish/v1/Managers/iDRAC.Embedded.1/Attributes

Header: content-type application/json

Auth: Basic or X auth

Body: {"Attributes": {"Lockdown.1.SystemLockdown": "<allowable values: Enabled
or Disabled>"}}
```

3 Section 3: System/Overview Page

1. Summary

Note: See the Dashboard page section 2, numbers 1 through 13 to get this information.

2. Batteries/System Board CMOS Battery Status

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellSensors/iDRAC.Embedded.1_0x23_
SystemBoardCMOSBattery?$select=HealthState

Header: content-type application/json

Auth: Basic or X auth
```

3. Batteries/PERC ROMB Battery Status

```
Command: GET

redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellSensors/iDRAC.Embedded.1_0x23_
PERC1ROMBBattery?$select=HealthState

Header: content-type application/json

Auth: Basic or X auth
```

4. Cooling/Cooling Overview/Fan Status

```
Command: GET

URI:
/redfish/v1/Chassis/System.Embedded.1/ThermalSubsystem/Fans?$expand=*($levels=1)

Header: content-type application/json
```

```
Auth: Basic or X auth
```

Note: Run GET on URI "/redfish/v1/Systems/System.Embedded.1?$select=Oem.Dell.DellSystem.FanRollupStatus" to get overall fan health.

5. Cooling/Cooling Overview/Fan Status/Fan Type

```
Command: GET

URI: /redfish/v1/Chassis/System.Embedded.1/ThermalSubsystem/Fans/{Fan component ID}?$select=Oem/Dell/FanType

Header: content-type application/json
```

6. Cooling/Cooling Overview/Redundancy Status

```
Command: GET

URI:
/redfish/v1/Chassis/System.Embedded.1/ThermalSubsystem?$select=FanRedundancy

Header: content-type application/json
```

NOTE: It is derived based on Status->State and Status->Health properties.

7. Cooling/Cooling Overview/Average Fan Speed

```
Command: GET

URI:
/redfish/v1/Chassis/System.Embedded.1/EnvironmentMetrics?$select=Oem/Dell/AverageFanPWM

Header: content-type application/json
```

8. Cooling/Cooling Overview/Net System Airflow

```
Command: GET

URI: /redfish/v1/Chassis/System.Embedded.1/Sensors/SystemAirFlow?$select=Reading

Header: content-type application/json
```

9. Cooling/Cooling Overview/Thermal Profile Optimization

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.ThermalProfile

Header: content-type application/json

Auth: Basic or X auth
```

10. Cooling/Cooling Overview/Fan Speed Offset

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.FanSpeedOffset

Header: content-type application/json

Auth: Basic or X auth
```

11. Cooling/Cooling Overview/Minimum Fan Speed

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.MFSMinimumLimit

Header: content-type application/json

Auth: Basic or X auth
```

12. Cooling/Cooling Overview/PCIe Airflow Settings

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/PCIeSlotLFM.<slot number>.LFMMode

URI Example:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/PCIeSlotLFM.2.LFMMode

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: There is no direct single property in Redfish to support this GUI property, it is derived based on "LFMMode" attribute for each PCIe card.

13. Cooling/Temperature Overview/Temperature Status

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded.
1_0x23_SystemBoardInletTemp

Header: content-type application/json

Auth: Basic or X auth
```

14. Cooling/Temperature Overview/System Inlet Temperature

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded.
1_0x23_SystemBoardInletTemp
```

```
Header: content-type application/json

Auth: Basic or X auth
```

15. Cooling/Temperature Overview/System Inlet Temperature Support Limit For This Configuration

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.SystemInletTemperatureSupportLimitPerConfigu
ration

Header: content-type application/json

Auth: Basic or X auth
```

16. Cooling/Temperature Overview/ASHRAE Category

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalConfig.1.ASHRAEEnvironmentalClass

Header: content-type application/json

Auth: Basic or X auth
```

17. Cooling/Fans Status

```
Command: GET

URI:
/redfish/v1/Chassis/System.Embedded.1/ThermalSubsystem/Fans?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

18. Cooling/Temperatures/Temperature Probes/CPU1 Temp

```
Command: GET

URI:
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded
.1_0x23_CPU1Temp

Header: content-type application/json

Auth: Basic or X auth
```

Note: This URI is only for CPU 1, if your config has other CPUs, change the component ID to the CPU socket. Example URI for CPU socket 2: “redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded.1_0x23_CPU2Temp”.

19. Cooling/Temperatures/Temperature Probes/System Board Exhaust

```
Command: GET

URI:
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded
.1_0x23_SystemBoardExhaustTemp

Header: content-type application/json

Auth: Basic or X auth
```

20. Cooling/Temperatures/Temperature Probes/System Board Inlet Temp

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded.
1_0x23_SystemBoardInletTemp

Header: content-type application/json
```

```
Auth: Basic or X auth
```

21. Cooling/Temperatures/Temperature Probes/Set Minimum Threshold (LowerCaution) System Board Inlet Temp

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardInletTemp

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds": {"LowerCaution": {"Reading": <pass in integer value>}}}
```

22. Cooling/Temperatures/Temperature Probes/Set Maximum Threshold (UpperCaution) System Board Inlet Temp

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardInletTemp

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds": {"UpperCaution": {"Reading": <pass in integer value>}}}
```

23. Cooling/Temperatures/Temperature Probes/Maximum DIMM Temperature

```
Command:  GET

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/Temperature.DIMM_MAX

Header: content-type application/json

Auth: Basic or X auth
```

24. CPU/CPUs/Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

25. Front Panel/Live Front Panel Feed

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/LCD.1.CurrentDisplay

Header: content-type application/json

Auth: Basic or X auth
```

26. Accelerators/GPUs/Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU ID}?$select=Status

Header: content-type application/json

Auth: Basic or X auth
```

Note: Run GET on URI “redfish/v1/Systems/System.Embedded.1/Processors” to get the GPU ID (Example: Video.Slot.1-1)

27. Accelerators/GPUs/Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU ID}?$select=Name
```

```
Header: content-type application/json

Auth: Basic or X auth
```

28. Accelerators/GPUs/Product Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU ID}?$select=Model

Header: content-type application/json

Auth: Basic or X auth
```

29. Accelerators/GPUs/Slot Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU ID}?$select=Id

Header: content-type application/json

Auth: Basic or X auth
```

Note: The first integer that is listed in the string value is the slot number. Example: "Video.Slot.1-1", card is in PCI slot 1.

30. Accelerators/GPUs/Board Part Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU ID}?$select=PartNumber

Header: content-type application/json

Auth: Basic or X auth
```

31. Accelerators/GPUs/Serial Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU
ID}?$select=SerialNumber

Header: content-type application/json

Auth: Basic or X auth
```

32. Accelerators/GPUs/ GPU Part Number

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portHWInventory

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType":"Local"}
```

Note: This command exports all system hardware information. In the response headers location, URI return is “/redfish/v1/Dell/hwinv.xml”. Run GET on this URI and parse the output, look for your GPU ID (Example: Video.Slot.1-1) which has details about this property.

33. Accelerators/GPUs/ GPU Firmware Version

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portHWInventory

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType":"Local"}
```

Note: This command exports all system hardware information. In the response headers location, URI return is “/redfish/v1/Dell/hwinv.xml”. Run GET on this URI and parse the output, look for your GPU ID (Example: Video.Slot.1-1) which has details about this property.

34. Accelerators/GPUs/Power

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU
ID}/Oem/Dell/PowerMetrics

Header: content-type application/json

Auth: Basic or X auth
```

35. Accelerators/GPUs/Temperature and Thermal

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{GPU
ID}/Oem/Dell/ThermalMetrics

Header: content-type application/json

Auth: Basic or X auth
```

36. Accelerators/FPGAs/Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA ID}?$select=Status

Header: content-type application/json

Auth: Basic or X auth
```

Note: Run GET on URI “redfish/v1/Systems/System.Embedded.1/Processors” to get the FPGA ID (Example: ProcAccelerator.Slot.1-1)

37. Accelerators/FPGAs/Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA ID}?$select=Name
```

```
Header: content-type application/json

Auth: Basic or X auth
```

38. Accelerators/FPGAs/Product Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA ID}?$select=Model

Header: content-type application/json

Auth: Basic or X auth
```

39. Accelerators/FPGAs/Slot Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA ID}?$select=Id

Header: content-type application/json

Auth: Basic or X auth
```

Note: The first integer that is listed in the string value is the slot number. Example: "ProcAccelerator.Slot.1-1", card is in PCI slot 1.

40. Accelerators/FPGAs/Board Part Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA
ID}?$select=PartNumber

Header: content-type application/json

Auth: Basic or X auth
```

41. Accelerators/FPGAs/Serial Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA
ID}?$select=SerialNumber

Header: content-type application/json

Auth: Basic or X auth
```

42. Accelerators/FPGAs/FPGA Part Number

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portHWInventory

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType":"Local"}
```

Note: This command exports all system hardware information. In the response headers location, URI return is “/redfish/v1/Dell/hwinv.xml”. Run GET on this URI and parse the output, look for your FPGA ID (Example: ProcAccelerator.Slot.1-1) which has details about this property.

43. Accelerators/FPGAs/FPGA Firmware Version

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portHWInventory

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType":"Local"}
```

Note: This command exports all system hardware information. In the response headers location, URI return is “/redfish/v1/Dell/hwinv.xml”. Run GET on this URI and parse the output, look for your GPU ID (Example: ProcAccelerator.Slot.1-1) which has details about this property.

44. Accelerators/FPGAs/Temperature

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Processors/{FPGA
ID}/FPGA/Oem/Dell/ThermalMetrics

Header: content-type application/json

Auth: Basic or X auth
```

45. Intrusion/Status

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1?$select=PhysicalSecurity/IntrusionSensor

Header: content-type application/json

Auth: Basic or X auth
```

46. Memory/Memory Attributes/Installed Capacity

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=MemorySummary/TotalSystemMemoryGiB

Header: content-type application/json

Auth: Basic or X auth
```

47. Memory/Memory Attributes/Maximum Capacity

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/MaxSystemMemory
MiB

Header: content-type application/json
```

```
Auth: Basic or X auth
```

48. Memory/Memory Attributes/Slots Available

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/MaxDIMMSlots

Header: content-type application/json

Auth: Basic or X auth
```

49. Memory/Memory Attributes/Slots Used

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/PopulatedDIMMSl
ots

Header: content-type application/json

Auth: Basic or X auth
```

50. Memory/Memory Attributes/Error Correction

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/SysMemErrorMeth
odology

Header: content-type application/json

Auth: Basic or X auth
```

51. Memory/Individual Memory Details

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Memory?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

52. Removable Media/Internal SD Module(Redundancy Status)

```
Command: GET

URI:
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellSystem/System.Embedded.1?$sel
ect=IDSDMRedundancyStatus

Header: content-type application/json

Auth: Basic or X auth
```

53. Removable Media/Internal SD Module Status

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellvFlash,
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellPresenceAndStatusSensors

Header: content-type application/json

Auth: Basic or X auth
```

54. Voltages/Voltages

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/Sensors?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

55. Network Devices/Network Devices/Summary

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters

Header: content-type application/json

Auth: Basic or X auth
```

Note: This command returns all NID IDs detected for the server (Example: NIC.Embedded.1)

56. Network Devices/Network Devices/Summary/NIC Devices/Status

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}?$select=Status

Header: content-type application/json

Auth: Basic or X auth
```

57. Network Devices/Network Devices/Summary/NIC Devices/Name

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapter?$select=Id

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To get the NIC port ID, run GET on URI "redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC ID}/NetworkDeviceFunctions"

58. Network Devices/Network Devices/Nic Devices/{NIC ID}/Product Name

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapter/{NIC
ID}/NetworkDeviceFunctions/{NIC Port ID}?$select=Oem/Dell/DellNIC/ProductName

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To get the NIC port ID, run GET on URI "redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC ID}/NetworkDeviceFunctions"

59. Network Devices/Network Devices/Summary/NIC Devices/CPU Affinity

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Embedded.1?$select=Cont
rollers

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To see the property CPUAffinity, parse the JSON data under Dell/Oem collection.

60. Network Devices/Network Devices/{NIC ID}/Product Name

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapter/{NIC
ID}/NetworkDeviceFunctions/{NIC Port ID}?$select=Oem/Dell/DellNIC/ProductName

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To get the NIC port ID, run GET on URI "redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC ID}/NetworkDeviceFunctions"

61. Network Devices/Network Devices/{NIC ID}/Vendor Name

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapter/{NIC
ID}/NetworkDeviceFunctions/{NIC Port ID}?$select=Oem/Dell/DellNIC/VendorName

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To get the NIC port ID, run GET on URI "redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC ID}/NetworkDeviceFunctions"

62. Network Devices/Network Devices/{NIC ID}/Number of Ports

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}?$select=Controllers

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To view the property NetworkPortCount, parse the JSON data under ControllerCapabilities collection.

63. Network Devices/Network Devices/{NIC ID}/SNAPI Support

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapter/{NIC
ID}/NetworkDeviceFunctions/{NIC Port ID}?$select=Oem/Dell/DellNIC/SNAPISupport

Header: content-type application/json

Auth: Basic or X auth
```

64. Network Devices/Network Devices/{NIC ID}/VPI Support

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapter/{NIC
ID}/NetworkDeviceFunctions/{NIC Port ID}?$select=Oem/Dell/DellNIC/VPISupport

Header: content-type application/json

Auth: Basic or X auth
```

65. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Link Status

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkPorts/{NIC port ID}?$select=LinkStatus

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To get NIC port IDs, run GET on URI "redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Embedded.1", check Controllers/Links/NetworkPorts collection.

66. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Link Speed

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkPorts/{NIC port ID}?$select=CurrentLinkSpeedMbps

Header: content-type application/json

Auth: Basic or X auth
```

67. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Protocol

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{NIC port ID}?$select=Oem/Dell/DellNIC/Protocol

Header: content-type application/json

Auth: Basic or X auth
```

68. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Switch Connection ID

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/NetworkPorts/Oem/Dell/DellSwitchConnections

Header: content-type application/json

Auth: Basic or X auth
```

Note: Parse the JSON output to get SwitchConnectionID property value for the NIC ID.

69. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Switch Port Connection ID

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/NetworkPorts/Oem/Dell/DellSwitchConnections

Header: content-type application/json

Auth: Basic or X auth
```

Note: To get the SwitchPortConnectionID property value for NIC ID, parse the JSON output.

70. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/CPU Affinity

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{NIC port ID}?$select=Links/Oem/Dell/CPUAffinity

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: If an empty collection is returned, that means this feature is not supported.

71. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Number of Partitions Supported

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{port number}/Oem/Dell/DellNetworkAttributes/{port
Number}?$select=Attributes/NumberPCIFunctionsSupported

URI Example:
redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDev
iceFunctions/NIC.Integrated.1-1-
1/Oem/Dell/DellNetworkAttributes/NIC.Integrated.1-1-
1?$select=Attributes/NumberPCIFunctionsSupported

Header: content-type application/json

Auth: Basic or X auth
```

72. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Number of Partitions Enabled

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{port number}/Oem/Dell/DellNetworkAttributes/{port
Number}?$select=Attributes/NumberPCIFunctionsEnabled
```

```
URI Example:
redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDev
iceFunctions/NIC.Integrated.1-1-
1/Oem/Dell/DellNetworkAttributes/NIC.Integrated.1-1-
1?$select=Attributes/NumberPCIFunctionsEnabled

Header: content-type application/json

Auth: Basic or X auth
```

73. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Family Firmware Version

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Embedded.1/NetworkDevic
eFunctions/NIC.Embedded.1-1-1?$select=Oem/Dell/DellNIC/FamilyVersion

Header: content-type application/json

Auth: Basic or X auth
```

74. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Auto Negotiation

```
Command: GET

URI redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkPorts/{NIC port}?$select=SupportedLinkCapabilities

URI Example:
redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkPor
ts/NIC.Integrated.1-1?$select=SupportedLinkCapabilities

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: In the JSON output, parse for property AutoSpeedNegotiation, will have a Boolean value of true or false.

75. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Settings and Capabilities

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkPorts/{NIC Port ID}

URI: Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{NIC port ID}/Oem/Dell/DellNICCapabilities/{NIC Port
ID}

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{NIC port ID}/Oem/Dell/DellNetworkAttributes/{NIC
port ID}

Header: content-type application/json

Auth: Basic or X auth
```

76. Network Devices/Network Devices/Nic Devices/{NIC ID}/{Port Number}/Port Statistics

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkDeviceFunctions/{NIC port ID/Oem/Dell/DellNICPortMetrics/{NIC port
ID}

Header: content-type application/json

Auth: Basic or X auth
```

77. Power/Power Supplies/Health/Name/Status/Input Wattage/Output Wattage/FW Version/Part Number/Input Line Type/Type

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/Power#/PowerSupplies (or)
/redfish/v1/Chassis/System.Embedded.1/PowerSubsystem/PowerSupplies?$expand=*

Header: content-type application/json
```

```
Auth: Basic or X auth
```

NOTE: Query parameters are not supported for the first URI, parse the JSON output to get these property values.

78. Power/Power/Last Hour Readings

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/Power#/PowerSupplies

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Query parameters are not supported for this URI. Parse the JSON output and look for PowerMetrics object to get last hour readings.

79. Power/Power/Last Day Readings

```
Command: GET

URI: redfish/v1/TelemetryService/MetricReports/PowerStatistics

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Query parameters are not supported for this URI, parse the JSON output, and look for the property “LastDayAvgPower”.

NOTE: The iDRAC Datacenter license is required to GET this data.

80. Power/Power/Last Week Readings

```
Command: GET

URI: redfish/v1/TelemetryService/MetricReports/PowerStatistics

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Query parameters are not supported for this URI, parse the JSON output, and look for property “LastWeekAvgPower”.

NOTE: The iDRAC Datacenter license is required to GET this data.

81. Power/Present Power Readings and Thresholds/Present Reading

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardPwrConsumption?$select=Reading

Header: content-type application/json

Auth: Basic or X auth
```

82. Power/Present Power Readings and Thresholds/Warning Threshold

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardPwrConsumption?$select=Thresholds/UpperCaution

Header: content-type application/json

Auth: Basic or X auth
```

83. Power/Present Power Readings and Thresholds/Failure Threshold

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardPwrConsumption?$select=Thresholds/UpperCritical

Header: content-type application/json
```

```
Auth: Basic or X auth
```

84. Power/Present Power Readings and Thresholds/Edit Warning Threshold

```
Command: PATCH

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardPwrConsumption?$select=T
hresholds/UpperCritical

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds":{"UpperCaution":{"Reading":<Integer value>}}}
```

85. Power/Power Supply Unit Readings/Amps

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/{PSU
Name}Current1?$select=Reading

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: PSU name is either PSU1, PSU2, PSU3, or PSU4.

86. Power/Power Supply Unit Readings/Volts

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellNumericSensors/iDRAC.Embedded.
1_0x23_{PSU Name}Voltage?$select=CurrentReading

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: PSU name is either PSU1, PSU2, PSU3, or PSU4.

87. Power/Raw Power Consumption

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/Power

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Query parameters are not supported for this URI, parse the JSON output, and look for PowerControl object, property PowerConsumedWatts returns this data.

88. Power/Cumulative Reading/Time

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwrMon.1.CumulativePowerStartTimeStr

Header: content-type application/json

Auth: Basic or X auth
```

89. Power/Cumulative Reading/Total Usage

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwrMon.1.AccumulativePower

Header: content-type application/json

Auth: Basic or X auth
```

90. Power/Historical Peaks/Time

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwrMon.1.PeakPowerStartTimeStr

Header: content-type application/json

Auth: Basic or X auth
```

91. Power/Historical Peaks/Peak Watts

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwrMon.1.PeakPowerWatts

Header: content-type application/json

Auth: Basic or X auth
```

92. Power/Historical Peaks/Peak Watts Time

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwrMon.1.PeakPowerTimeStr

Header: content-type application/json

Auth: Basic or X auth
```

93. Power/Historical Peaks/Peak Amps

NOTE: Not supported by Redfish

94. Power/Historical Peaks/Peak Amps Time

NOTE: Not supported by Redfish

95. Power/System Headroom/Instantaneous

NOTE: Not supported by Redfish

96. Power/System Headroom/Peak

NOTE: Not supported by Redfish

97. PCIe Slots/PCIe Slot Details

```
Command: GET

URI: redfish/v1/Dell/Systems/System.Embedded.1/DellSlotCollection

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: For all PCIe slot entries, parse the JSON output and look for “ConnectorLayout” with value “PCI-E”.

NOTE: Depending on how many slots the server supports, use query parameter skip, to see all slot devices (Example: redfish/v1/Dell/Systems/System.Embedded.1/DellSlotCollection?$skip=50).

98. PCIe Slots/PCIe Slot Details/PCIeDevices

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/PCIeDevices/{bus device ID}

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: To get the bus device ID, run GET on URI “redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC ID}/NetworkDeviceFunctions/{NIC port ID}?$select=Oem/Dell/DellNIC/BusNumber”.

Example URI:

redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Integrated.1/NetworkDeviceFunctions/NIC.Integrated.1-1-1?$select=Oem/Dell/DellNIC/BusNumber

Bus ID value returned should be a digit value (Example: 177), now append “-0” to create the bus ID for the URI.

Example URI:

redfish/v1/Systems/System.Embedded.1/PCIeDevices/177-0

4 Section 4: System/Details Page

1. System Details/Overview/Health

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=Status

Header: content-type application/json

Auth: Basic or X token
```

2. System Details/Overview/Power State

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=PowerState

Header: content-type application/json

Auth: Basic or X auth
```

3. System Details/Overview/Model

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=Model

Header: content-type application/json

Auth: Basic or X auth
```

4. System Details/Overview/Host Name

```
Command: GET
```

```
URI: redfish/v1/Systems/System.Embedded.1?$select=HostName

Header: content-type application/json

Auth: Basic or X auth
```

5. System Details/Overview/Operating System

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerOS.1.OSName

Header: content-type application/json

Auth: Basic or X auth
```

6. System Details/Overview/Operating System Version

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerOS.1.OSVersion

Header: content-type application/json

Auth: Basic or X auth
```

7. System Details/Overview/Service Tag

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/Bios?$select=Attributes/SystemServiceTag

Header: content-type application/json

Auth: Basic or X auth
```

8. System Details/Overview/Asset Tag

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Bios?$select=Attributes/AssetTag

Header: content-type application/json

Auth: Basic or X auth
```

9. System Details/Overview/Express Service Code

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/ExpressServiceC
ode

Header: content-type application/json

Auth: Basic or X auth
```

10. System Details/Overview/BIOS Version

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=BiosVersion

Header: content-type application/json

Auth: Basic or X auth
```

11. System Details/Overview/Lifecycle Controller

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1?$select=FirmwareVersion

Header: content-type application/json
```

```
Auth: Basic or X auth
```

NOTE: iDRAC and Lifecycle Controller will report the same version.

12. System Details/Overview/System Revision

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Oem/Dell/DellSystem/SystemRevision

Header: content-type application/json

Auth: Basic or X auth
```

13. System Details/Overview/IDSDM Firmware Version

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Parse the JSON output and look for URI with naming “internal.dualsdmodule”, to locate the installed version for IDSDM.

Example: "/redfish/v1/UpdateService/FirmwareInventory/Installed-103710-1.7__internal.dualsdmodule.1"

14. System Details/Overview/Get Location Attributes

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/{Location attribute name}

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: For location attributes, here are the supported attribute names: ServerTopology.1.DataCenterName, ServerTopology.1.AisleName, ServerTopology.1.RackName, ServerTopology.1.RoomName, and ServerTopology.1.RackSlot

15. System Details/Overview/Set Location Attributes

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/{Location attribute name}

Header: content-type application/json

Auth: Basic or X auth

Body: {"Attributes":{"<location attribute name>":"<location attribute value>"}}
```

NOTE: For location attributes, supported attribute names: ServerTopology.1.DataCenterName, ServerTopology.1.AisleName, ServerTopology.1.RackName, ServerTopology.1.RoomName, and ServerTopology.1.RackSlot

16. System Details/Auto Recover

NOTE: Not supported by Redfish

17. System Details/NIC MAC Addresses

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/{NIC
ID}/NetworkPorts?$expand=*($levels=1)

URI Example:
redfish/v1/Chassis/System.Embedded.1/NetworkAdapters/NIC.Embedded.1/NetworkPorts
?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Parse the JSON output and object AssociatedNetworkAddresses reports the MAC address.

18. iDRAC Details/RAC Information/Name

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1?$select=Oem/Dell/DelliDRACCard/Name

Header: content-type application/json

Auth: Basic or X auth
```

19. iDRAC Details/RAC Information/License

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellLicenses

Header: content-type application/json

Auth: Basic or X auth
```

20. iDRAC Details/RAC Information/Date/Time

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1?$select=DateTime

Header: content-type application/json

Auth: Basic or X auth
```

21. iDRAC Details/RAC Information/Firmware Version

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1?$select=FirmwareVersion
```

```
Header: content-type application/json

Auth: Basic or X auth
```

22. iDRAC Details/RAC Information/Firmware Updated

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory/Installed-25227-
6.00.02.00__iDRAC.Embedded.1-
1?$select=Oem/Dell/DellSoftwareInventory/InstallationDate

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Replace “6.00.02.00” with your iDRAC version in the URI.

23. iDRAC Details/RAC Information/Hardware Version

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Info.1.HWRev

Header: content-type application/json

Auth: Basic or X auth
```

24. iDRAC Details/RAC Information/MAC Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.MACAddress

Header: content-type application/json

Auth: Basic or X auth
```

25. iDRAC Details/RAC Information/DNS Domain Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ NIC.1.DNSDomainName

Header: content-type application/json

Auth: Basic or X auth
```

26. iDRAC Details/RAC Information/Use DHCP to Obtain DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NICStatic.1.DNSDomainFromDHCP

Header: content-type application/json

Auth: Basic or X auth
```

27. iDRAC Details/IPv4 Information/IPv4 Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

28. iDRAC Details/IPv4 Information/Current IP Address

```
Command: GET
```

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/CurrentIPv4.1.Address

Header: content-type application/json

Auth: Basic or X auth

29. iDRAC Details/IPv4 Information/Current Subnet Mask

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/CurrentIPv4.1.Netmask

Header: content-type application/json

Auth: Basic or X auth

30. iDRAC Details/IPv4 Information/Current Gateway

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/CurrentIPv4.1.Gateway

Header: content-type application/json

Auth: Basic or X auth

31. iDRAC Details/IPv6 Information/IPv6 Enabled

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/IPv6.1.Enable

Header: content-type application/json

```
Auth: Basic or X auth
```

32. iDRAC Details/IPv6 Information/Current IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.Address1

Header: content-type application/json

Auth: Basic or X auth
```

33. iDRAC Details/IPv6 Information/Current IP Gateway

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.Gateway

Header: content-type application/json

Auth: Basic or X auth
```

34. iDRAC Details/Ipv6 Information/Link Local Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.LinkLocalAddress

Header: content-type application/json

Auth: Basic or X auth
```

35. iDRAC Details/IPv6 Information/Autoconfiguration

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.AutoConfig

Header: content-type application/json

Auth: Basic or X auth
```

36. iDRAC Details/IPv6 Information/Use DHCPv6 to Obtain DNS Server Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.DNSFromDHCP6

Header: content-type application/json

Auth: Basic or X auth
```

37. DRAC Details/Ipv6 Information/Current Preferred DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.DNS1

Header: content-type application/json

Auth: Basic or X auth
```

38. iDRAC Details/Ipv6 Information/Current Alternate DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.DNS2

Header: content-type application/json

Auth: Basic or X auth
```

39. Asset Tracking/System Information/System Boot Time

```
Command: GET

URI:
redfish/v1/Managers/System.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?
$select=Attributes/SystemInfo.1.BootTime

Header: content-type application/json

Auth: Basic or X auth
```

40. Asset Tracking/System Information/System Time

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1?$select=DateTime

Header: content-type application/json

Auth: Basic or X auth
```

5 Section 5: System/Inventory Page

1. Firmware Inventory

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X token
```

2. Hardware Inventory

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X token
```

NOTE: This command does not expand all sub URIs to display hardware details. Recommended using OEM action DellLCService.ExportHWInventory. This command gets all server hardware information in XML format which this XML file can be exported locally (example below) or to a network share.

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.ExportHWInventory

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType":"Local"}
```

NOTE: In the headers location, see the URI returned (/redfish/v1/Dell/hwinv.xml). Run GET on this URI to download hardware inventory file.

6 Section 6: System/Performance Page

1. CPU Usage

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardCPUUsage?$select=Reading

Header: content-type application/json

Auth: Basic or X token
```

2. CPU Warning Threshold Setting

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardCPUUsage

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds":{"UpperCaution":{"Reading":<integer value>}}}
```

3. CPU Last Hour/Day/Week Readings

NOTE: Supported but only using Telemetry custom metric reports which requires iDRAC Datacenter license.

4. Historical Peak For System Board CPU Usage

NOTE: Not supported by Redfish

5. Memory Usage

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardMEMUsage?$select=Reading

Header: content-type application/json

Auth: Basic or X token
```

6. Memory Warning Threshold Setting

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardMEMUsage

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds":{"UpperCaution":{"Reading":<integer value>}}}
```

7. Memory Last Hour/Day/Week Readings

NOTE: Supported but only using Telemetry custom metric reports which requires iDRAC Datacenter license.

8. Historical Peak For System Board Memory Usage

NOTE: Not supported by Redfish

9. I/O Usage

```
Command: GET
```

```
URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardIOUsage?$select=Reading

Header: content-type application/json

Auth: Basic or X token
```

10. I/O Warning Threshold Setting

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardIOUsage

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds":{"UpperCaution":{"Reading":<integer value>}}}
```

11. I/O Last Hour/Day/Week Readings

NOTE: Supported but only using Telemetry custom metric reports which requires iDRAC Datacenter license.

12. Historical Peak For System Board I/O Usage

NOTE: Not supported by Redfish

13. System Usage

```
Command: GET

URI:
redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardSYSUsage?$select=Reading

Header: content-type application/json

Auth: Basic or X token
```

14. System Warning Threshold Setting

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1/Sensors/SystemBoardSYSUsage

Header: content-type application/json

Auth: Basic or X auth

Body: {"Thresholds":{"UpperCaution":{"Reading":<integer value>}}}
```

15. System Last Hour/Day/Week Readings

NOTE: Supported but only using Telemetry custom metric reports which requires iDRAC Datacenter license.

16. Historical Peak For System Board System Usage

NOTE: Not supported by Redfish

7 Section 7: System/Host operating system

NOTE: For information about the system or host operating system, ensure that the iDRAC Service Module (iSM) is installed and running in the operating system.

1. Check iSM installed, and service is running in the operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.ServiceModuleState

Header: content-type application/json

Auth: Basic or X token
```

NOTE: Value of Running returned means iSM is installed and service is running.

2. Network Interfaces

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1/EthernetInterfaces?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X token
```

8 Section 8: Storage/Overview/Controllers

1. Controllers

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage

Header: content-type application/json

Auth: Basic or X token
```

2. Controller/Rollup Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Oem/Dell/DellController/RollupStatus

Header: content-type application/json

Auth: Basic or X token
```

3. Controller/Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-1?$select=Name

Header: content-type application/json

Auth: Basic or X token
```

4. Controller/Device Description

```
Command: GET
```

```
URI: redfish/v1/Systems/System.Embedded.1/Storage/RAID.SL.3-
1?$select=Description

Header: content-type application/json

Auth: Basic or X token
```

5. Controller/Firmware Version

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Oem/Dell/DellController/ControllerFirmwareVersion

Header: content-type application/json

Auth: Basic or X token
```

6. Controller/Driver Version

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Oem/Dell/DellController/DriverVersion

Header: content-type application/json

Auth: Basic or X token
```

7. Controller/Cache Memory Size

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Oem/Dell/DellController/CacheSizeInMB

Header: content-type application/json

Auth: Basic or X token
```

8. Controller/Controller Properties/Security/Rates

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Oem/Dell/DellController

Header: content-type application/json

Auth: Basic or X token
```

9. Controller/PCIe

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=StorageControllers

Header: content-type application/json

Auth: Basic or X token
```

NOTE: In the output look for Links/PCIeFunctions, @odata.id property should report URI to get this data (Example: /redfish/v1/Systems/System.Embedded.1/PCIeDevices/101-0/PCIeFunctions/101-0-0). Run GET on this URI to get PCIe details for the storage controller.

10. Controller/Controller Battery

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Oem/Dell/DellControllerBattery

Header: content-type application/json

Auth: Basic or X token
```

11. Reset Storage Controller

```
Command: POST
```

```
URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.ResetConfig

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD": "<controller ID>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

12. Clear Foreign Configuration

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.ClearForeignConfig

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD": "<controller ID>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

13. Import Foreign Configuration

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.ClearForeignConfig

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD": "<controller ID>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

14. Set Controller Key (Local Key Management)

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.SetControllerKey

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<controller ID>","Key":"<new passphrase>","Keyid":"<new key
ID>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

15. Rekey Controller Key (Local Key Management)

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.ReKey

Header: content-type application/json

Auth: Basic or X token

Body: {"Mode":"LKM","TargetFQDD":"<controller ID>","OldKey":"<current key
phrase>","NewKey":"<new key passphrase>","Keyid":"<pass in current or new key ID
to change>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

16. Remove Controller Key (Local Key Management)

```
Command: POST
```

```
URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.RemoveControllerKey

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD": "<controller ID>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

9 Section 9: Storage/Overview/Physical Disks

9.1 Physical Disks

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}?$select=Drives

Header: content-type application/json

Auth: Basic or X token
```

9.2 Physical Disk/Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=Status

Header: content-type application/json

Auth: Basic or X token
```

9.3 Physical Disk/Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=Name

Header: content-type application/json

Auth: Basic or X token
```

9.4 Physical Disk/State

```
Command: GET
```

```
URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=Oem/Dell/DellPhysicalDisk/RaidStatus

Header: content-type application/json

Auth: Basic or X token
```

9.5 Physical Disk/Slot Number

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=Oem/Dell/DellPhysicalDisk/Slot

Header: content-type application/json

Auth: Basic or X token
```

9.6 Physical Disk/Size

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=CapacityBytes

Header: content-type application/json

Auth: Basic or X token
```

9.7 Physical Disk/Bus Protocol

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=Protocol

Header: content-type application/json

Auth: Basic or X token
```

9.8 Physical Disk/Media Type

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=MediaType

Header: content-type application/json

Auth: Basic or X token
```

9.9 Physical Disk/Hot Spare

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}?$select=HotspareType

Header: content-type application/json

Auth: Basic or X token
```

9.10 Physical Disk/Drive Details/RAID Information/Security/Manufacturer Information

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{disk
ID}

Header: content-type application/json

Auth: Basic or X token
```

9.11 Blink Drive

```
Command: POST
```

```
URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.BlinkTarget

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<disk ID>"}
```

9.12 Unblink Drive

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.UnBlinkTarget

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<disk ID>"}
```

9.13 Cryptographic Erase Drive

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Drives/{drive
ID}/Actions/Drive.SecureErase

Header: content-type application/json

Auth: Basic or X token

Body: {}
```

NOTE: You must pass in empty body.

NOTE: In the Headers output, Location property returns a job ID URI, run GET on this URI to monitor the job status until it is marked as completed. If the job stops in “Scheduled” state, a server reboot is required to run the job.

9.14 Assign Global HotSpare

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.AssignSpare

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<disk FQDD>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

9.15 Assign Dedicated HotSpare

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.AssignSpare

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<disk FQDD>","VirtualDiskArray":["<virtual disk ID(s)>"]}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

9.16 Unassign HotSpare

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.UnassignSpare

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"TargetFQDD":"<disk FQDD>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

10 Section 10: Storage/Overview/Virtual Disks

1. Virtual Disks

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Volumes

Header: content-type application/json

Auth: Basic or X token
```

2. Virtual Disk/Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=Status

Header: content-type application/json

Auth: Basic or X token
```

3. Virtual Disk/Name

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=Name

Header: content-type application/json

Auth: Basic or X token
```

4. Virtual Disk/State

```
Command: GET
```

```
URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=Oem/Dell/DellVirtualDisk/RaidStatus

Header: content-type application/json

Auth: Basic or X token
```

5. Virtual Disk/Layout

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=RAIDType

Header: content-type application/json

Auth: Basic or X token
```

6. Virtual Disk/Size

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=CapacityBytes

Header: content-type application/json

Auth: Basic or X token
```

7. Virtual Disk/Media Type

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=Oem/Dell/DellVirtualDisk/MediaType

Header: content-type application/json

Auth: Basic or X token
```

8. Virtual Disk/Write Policy

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=WriteCachePolicy

Header: content-type application/json

Auth: Basic or X token
```

9. Virtual Disk/Virtual Disk Details

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}/Volumes/{virtual disk ID}?$select=Oem/Dell/DellVirtualDisk

Header: content-type application/json

Auth: Basic or X token
```

10. Create Virtual Disk

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller ID}/Volumes

Header: content-type application/json

Auth: Basic or X token

Body: {"RAIDType": "<RAID level>", "Drives": [{"@odata.id": "<disk URI>"}]}

Body Example: {"RAIDType": "RAID0", "Drives": [{"@odata.id":
"/redfish/v1/Systems/System.Embedded.1/Storage/Drives/Disk.Bay.12:Enclosure.Exte
rnal.0-0:RAID.Slot.2-1"}]}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed. If the job stops in “Scheduled” state, a server reboot is required to run the job.

11. Initialize Virtual Disk

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Storage/Volumes/{virtual disk ID}

Header: content-type application/json

Auth: Basic or X token

Body: {"InitializeType":"<Init type value>"}
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed. If the job stops in “Scheduled” state, a server reboot is required to run the job.

12. Delete Virtual Disk

```
Command: DELETE

URI: redfish/v1/Systems/System.Embedded.1/Storage/Volumes/{virtual disk ID}

Header: content-type application/json

Auth: Basic or X token

Body: No body required
```

NOTE: In the Headers output, Location property returns job ID URI, run GET on this URI to monitor the job status until it is marked as completed. If the job stops in “Scheduled” state, a server reboot is required to run the job.

13. Rename Virtual Disk

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.RenameVD

Header: content-type application/json

Auth: Basic or X token

Body: {"Name":"<new VD name value>", "TargetFQDD":"<virtual disk ID>"}
```

NOTE: In the Headers output, Location property returns a job ID URI, run GET on this URI to monitor the job status until it is marked as completed.

14. Blink Virtual Disk

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.BlinkTarget

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<virtualdisk ID>"}
```

15. Unblink Virtual Disk

```
Command: POST

URI:
redfish/v1/Dell/Systems/System.Embedded.1/DellRaidService/Actions/DellRaidServic
e.UnBlinkTarget

Header: content-type application/json

Auth: Basic or X token

Body: {"TargetFQDD":"<virtualdisk ID>"}
```

11 Section 11: Storage/Overview/Enclosures

1. Enclosures

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Storage/{controller
ID}?$select=Links/Enclosures

Header: content-type application/json

Auth: Basic or X token
```

2. Enclosure/Status

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}?$select=Status

Header: content-type application/json

Auth: Basic or X token
```

3. Enclosure/Enclosure ID

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}?$select=Model

Header: content-type application/json

Auth: Basic or X token
```

4. Enclosure/Associated Controllers

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}?$select=Description
```

```
Header: content-type application/json

Auth: Basic or X token
```

5. Enclosure/State

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}?$select=Status

Header: content-type application/json

Auth: Basic or X token
```

6. Enclosure/Fans

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}/ThermalSubsystem/Fans?$expand=*

Header: content-type application/json

Auth: Basic or X token
```

7. Enclosure/Power Supplies

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}/PowerSubsystem/PowerSupplies?$expand=*

Header: content-type application/json

Auth: Basic or X token
```

8. Enclosure/Temperature Probes

```
Command: GET

URI: redfish/v1/Chassis/{enclosure ID}/Sensors?$expand=*

Header: content-type application/json
```

```
Auth: Basic or X token
```

9. Enclosure/EMM

```
Command: GET

URI: redfish/v1/Chassis/{enclosure
ID}?$select=Oem/Dell/DellChassisEnclosure/Links/DellEnclosureEMMCollection

Header: content-type application/json

Auth: Basic or X token
```

12 Section 12: Configuration/Power Management

1. Power Control

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "<allowable value>"}
```

Note: For supported allowable values, run GET on URI "redfish/v1/Systems/System.Embedded.1?$select=Actions" and view property "ResetType@Redfish.AllowableValues".

2. Power Cap Policy/Active Power Cap Policy

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.ActivePowerCapVal

Header: content-type application/json

Auth: Basic or X token
```

3. Power Cap Policy/Power Cap

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.PowerCapSetting

Header: content-type application/json

Auth: Basic or X token
```

4. Power Cap Policy/Current Cap Value

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.PowerCapValue

Header: content-type application/json

Auth: Basic or X token
```

5. Power Cap Policy/Power Cap Limit Min

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.PowerCapMinThres

Header: content-type application/json

Auth: Basic or X token
```

6. Power Cap Policy/Power Cap Limit Max

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.PowerCapMaxThres

Header: content-type application/json

Auth: Basic or X token
```

7. Power Cap Policy/Set Power Cap

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServerPwr.1.PowerCapSetting": "Enabled",
"ServerPwr.1.PowerCapValue":"<power cap value you want to set>"}}
```

8. Power Cap Policy/Disable Power Cap

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServerPwr.1.PowerCapSetting": "Disabled"}}
```

9. Power Configuration/Get Redundancy Policy

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.PSRedPolicy

Header: content-type application/json

Auth: Basic or X token
```

10. Power Configuration/Set Redundancy Policy

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1
```

```
Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServerPwr.1.PSRedPolicy": "<possible values: Not
Redundant and A/B Grid Redundant>"}}
```

11. Power Configuration/Get Hot Spare

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.PSRapidOn

Header: content-type application/json

Auth: Basic or X token
```

12. Power Configuration/Set Hot Spare

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServerPwr.1.PSRapidOn": "<pass in Enabled or Disabled>"}}
```

13. Power Configuration/Get Primary PSU

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ServerPwr.1.RapidOnPrimaryPSU

Header: content-type application/json
```

```
Auth: Basic or X token
```

14. Power Configuration/Set Primary PSU

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServerPwr.1.RapidOnPrimaryPSU": "<pass in PSU1, PSU2,
PSU3 or PSU4>"}}
```

13 Section 13: Configuration/Virtual Console/Virtual Console

1. Virtual Console/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.Enable

Header: content-type application/json

Auth: Basic or X token
```

2. Virtual Console/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualConsole.1.Enable": "<pass in Enabled or
Disabled>"}}
```

3. Virtual Console/Get Max Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.MaxSessions

Header: content-type application/json

Auth: Basic or X token
```

4. Virtual Console/Set Max Sessions

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualConsole.1.MaxSessions": "<pass in integer value 1
to 6>"}}
```

5. Virtual Console/Active Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.ActiveSessions

Header: content-type application/json

Auth: Basic or X token
```

6. Virtual Console/Video Encryption

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.EncryptEnable

Header: content-type application/json

Auth: Basic or X token
```

7. Virtual Console/Get Local Server Video

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.LocalVideo

Header: content-type application/json

Auth: Basic or X token
```

8. Virtual Console/Set Local Server Video

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualConsole.1.LocalVideo": "<pass in Enabled or
Disabled>"}}
```

9. Virtual Console/Get Dynamic Action on Sharing Request Timeout

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.AccessPrivilege

Header: content-type application/json

Auth: Basic or X token
```

10. Virtual Console/Set Dynamic Action on Sharing Request Timeout

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"VirtualConsole.1.AccessPrivilege": "<pass in Full Access,
Read Only Access or Deny Access>"}}
```

11. Virtual Console/Get Automatic System Lock

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/AutoOSLockGroup.1.AutoOSLockState

Header: content-type application/json

Auth: Basic or X token
```

12. Virtual Console/Set Automatic System Lock

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"AutoOSLockGroup.1.AutoOSLockState": "<pass in Enabled or
Disabled>"}}
```

13. Virtual Console/Get Keyboard Mouse Attach State

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualConsole.1.AttachState

Header: content-type application/json

Auth: Basic or X token
```

14. Virtual Console/Set Keyboard Mouse Attach State

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualConsole.1.AttachState": "<pass in Detached,
Attached or Auto-attach>"}}
```

15. Virtual Console/Launch Virtual Console

a. Export: Server SSL Cert

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRAC
CardService.ExportSSLCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"SSLCertType":"Server"}
```

NOTE: The JSON output “CertificateFile” value contains the cert data.

b. Get KVM Session Details, Temp Username, and Password

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DelliDRACCardService/Actions/Delli
DRACCardService.GetKVMSession

Auth: Basic or X token
```

```
Body: {"SessionTypeName":"<complete SSL cert string value>"}
```

NOTE: In the JSON output, temp username and password are returned. This information is used to launch virtual console session (run POST command below).

c. Launch Virtual Console Session

Using any browser, pass in this URI to launch the session.

https://<iDRAC IP>/console?username=<current iDRAC username>&tempUsername=<temp username>&tempPassword=<temp password>

14 Section 14: Configuration/Virtual Console/VNC Server

1. VNC Server/Get Enable VNC Server

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/VNCServer.1.Enable

Header: content-type application/json

Auth: Basic or X token

2. VNC Server/Set Enable VNC Server

Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VNCServer.1.Enable": "<pass in Enabled or Disabled>"}}

3. VNC Server/Get VNC Password

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/VNCServer.1.Password

Header: content-type application/json

Auth: Basic or X token

NOTE: Current value returns “None” even if the password is set.

4. VNC Server/Set VNC Password

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VNCServer.1.Password": "<new value you want to set>"}}
```

5. VNC Server/Get Max Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VNCServer.1.MaxSessions

Header: content-type application/json

Auth: Basic or X token
```

6. VNC Server/Set Max Sessions

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VNCServer.1.MaxSessions": <pass in integer values 1 to
2>}}
```

7. VNC Server/Get Active Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VNCServer.1.ActiveSessions

Header: content-type application/json

Auth: Basic or X token
```

8. VNC Server/Get VNC Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VNCServer.1.Port

Header: content-type application/json

Auth: Basic or X token
```

9. VNC Server/Set Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VNCServer.1.Port": <pass in integer value 1024 to
65535>}}
```

10. VNC Server/Get Timeout

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VNCServer.1.Timeout

Header: content-type application/json

Auth: Basic or X token
```

11. VNC Server/Set Timeout

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VNCServer.1.Timeout": <pass in integer value from 60 to
10800>}}
```

12. VNC Server/Get SSL Encryption

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VNCServer.1.SSLEncryptionBitLength

Header: content-type application/json

Auth: Basic or X token
```

13. VNC Server/Set SSL Encryption

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"VNCServer.1.SSLEncryptionBitLength": "<pass in Disabled,
Auto Negotiate, 128-Bit or higher, 168-Bit or higher or 256-Bit or higher>"}}
```

15 Section 15: Configuration/Virtual Media/Attach Media

1. Attached Media/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.Enable

Header: content-type application/json

Auth: Basic or X token
```

2. Attached Media/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualMedia.1.Enable": "<pass in Enabled or Disabled>"}}
```

3. Attached Media/Get Attach Mode

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.Attached

Header: content-type application/json

Auth: Basic or X token
```

4. Attached Media/Set Attach Mode

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualMedia.1.Attached": "<pass in Detached, Attached or
AutoAttach>"}}
```

5. Attached Media/Get Max Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.MaxSessions

Header: content-type application/json

Auth: Basic or X token
```

6. Attached Media/Get Active Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.ActiveSessions

Header: content-type application/json

Auth: Basic or X token
```

7. Attached Media/Get Virtual Media Encryption

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.EncryptEnable

Header: content-type application/json

Auth: Basic or X token
```

8. Attached Media/Get Floppy Emulation

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.FloppyEmulation

Header: content-type application/json

Auth: Basic or X token
```

9. Attached Media/Set Floppy Emulation

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualMedia.1.FloppyEmulation": "<pass in Enabled or
Disabled>"}}
```

10. Attached Media/Get Boot Once

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.BootOnce
```

```
Header: content-type application/json

Auth: Basic or X token
```

11. Attached Media/Set Boot Once

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"VirtualMedia.1.BootOnce": "<pass in Enabled or
Disabled>"}}
```

12. Attached Media/Get Connection Status

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/VirtualMedia.1.ActiveSessions

Header: content-type application/json

Auth: Basic or X token
```

NOTE: Use the attribute VirtualMedia.1.ActiveSessions to get connection status. If value returned is not 0, the virtual media is connected.

16 Section 16: Configuration/Virtual Media/Remote File Share 1

1. Remote File Share 1/Get Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/VirtualMedia/1

Header: content-type application/json

Auth: Basic or X token
```

2. Remote File Share 1/Attach Device

```
Command: POST

URI:
redfish/v1/Systems/System.Embedded.1/VirtualMedia/1/Actions/VirtualMedia.InsertM
edia

Header: content-type application/json

Auth: Basic or X token

Body: {"Image": "<complete URI path of the ISO or IMG file>", "Inserted": true,
"WriteProtected": true}


Body Example: {"Image": "192.168.0.130:/nfs/boot.iso", "Inserted": true,
"WriteProtected": true}
```

3. Remote File Share 1/Detach Device

```
Command: POST

URI:
redfish/v1/Systems/System.Embedded.1/VirtualMedia/1/Actions/VirtualMedia.EjectMe
dia

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {}
```

NOTE: Pass in empty body.

4. Remote File Share 2/Get Status

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/VirtualMedia/2

Header: content-type application/json

Auth: Basic or X token
```

5. Remote File Share 2/Attach Device

```
Command: POST

URI:
redfish/v1/Systems/System.Embedded.1/VirtualMedia/2/Actions/VirtualMedia.InsertM
edia

Header: content-type application/json

Auth: Basic or X token

Body: {"Image": "<complete URI path of the ISO or IMG file>", "Inserted": true,
"WriteProtected": true}

Body Example: {"Image": "192.168.0.130:/nfs/boot.iso", "Inserted": true,
"WriteProtected": true}
```

6. Remote File Share 2/Detach Device

```
Command: POST

URI:
redfish/v1/Systems/System.Embedded.1/VirtualMedia/2/Actions/VirtualMedia.EjectMe
dia

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {}
```

NOTE: Pass in empty body.

17 Section 17: Configuration/Licenses

1. Get Installed Licenses

```
Command: GET

URI: redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLicenseCollection

Header: content-type application/json

Auth: Basic or X token
```

2. Export License Locally

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLicenseManagementService/Actions/D
ellLicenseManagementService.ExportLicense

Header: content-type application/json

Auth: Basic or X token

Body: {"EntitlementID":"<license ID>"}
```

NOTE: License details are in the JSON output returned.

3. Export License Network Share

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLicenseManagementService/Actions/D
ellLicenseManagementService.ExportLicenseToNetworkShare

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"EntitlementID":"<license ID>","LicenseName":"<pass in unique license
filename>","IPAddress":"<network share IP>","ShareType":"<share type, example
NFS, or HTTP>","ShareName":"<name of your network share>"}
```

NOTE: In the Headers Location, job ID URI is returned. Run GET on this URI to check the job status for export operation.

4. Import License Locally

```
URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLicenseManagementService/Actions/D
ellLicenseManagementService.ImportLicense

Header: content-type application/json

Auth: Basic or X token

Body: {"FQDD":"iDRAC.Embedded.1","ImportOptions":"Force","LicenseFile":"<base64
license string>"}
```

NOTE: Import license locally, the license must be converted to base64 for the body.

5. Import License Network Share

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLicenseManagementService/Actions/D
ellLicenseManagementService.ImportLicenseFromNetworkShare

Header: content-type application/json

Auth: Basic or X token

Body: {"FQDD":"iDRAC.Embedded.1","ImportOptions":"Force","LicenseName":"<pass in
unique license filename>","IPAddress":"<network share IP>","ShareType":"<share
type value, example, NFS, or HTTP>","ShareName":"<name of your network share>"}
```

NOTE: In the Headers Location, job ID URI is returned. Run GET on this URI to check the job status for import operation.

6. Delete License

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLicenseManagementService/Actions/D
ellLicenseManagementService.DeleteLicense

Header: content-type application/json

Auth: Basic or X token

Body: {"EntitlementID":"<license ID>","DeleteOptions":"Force"}
```

18 Section 18: Configuration/System Settings/Alert Configuration

1. Alert Configuration/Alerts/Get Enabled Status

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMILan.1.AlertEnable

Header: content-type application/json

Auth: Basic or X token
```

2. Alert Configuration/Alerts/Set Enabled Status

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMILan.1.AlertEnable": "<pass in Enabled or Disabled>"}}
```

3. Alert Configuration/Alerts/Get Alert Configuration

NOTE: You must leverage iDRAC Server Configuration Profile (SCP) feature to get iDRAC alerts.

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ExportSystemCon
figuration

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"ExportFormat": "<pass in JSON or XML>", "ShareParameters": {"Target":
"EventFilters", "IPAddress": "<network share IP>", "ShareType": "<pass in share
type value, examples, NFS, or HTTP>", "ShareName": "<network share name>",
"FileName": "<pass in unique filename>"}}
```

NOTE: In the Headers Location, job ID URI is returned. Run GET on this URI to check the job status for import operation.

4. Alert Configuration/Alerts/Set Alert Configuration

NOTE: You must leverage iDRAC Server Configuration Profile (SCP) feature to set iDRAC alerts.

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ImportSystemCon
figuration

Header: content-type application/json

Auth: Basic or X token

Body: {"ShareParameters": {"Target": "EventFilters", "IPAddress": "<network
share IP>", "ShareType": "<pass in share type value, examples, NFS, or HTTP>",
"ShareName": "<network share name>", "FileName": "<pass in your SCP filename
which has been edited to make alert changes>"}}
```

NOTE: In the Headers Location, job ID URI is returned. Run GET on this URI to check the job status for import operation.

5. Alert Configuration/SNMP Traps Configuration/Get Alert Destination 1 State

NOTE: Get and set examples are for destination 1. If you want to configure a different destination, change the index number.

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMPAlert.1.State

Header: content-type application/json

Auth: Basic or X token
```

6. Alert Configuration/SNMP Traps Configuration/Set Alert Destination 1 State

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMPAlert.1.State": "<pass in Enabled or Disabled>"}}
```

7. Alert Configuration/SNMP Traps Configuration/Get Alert Destination 1 Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMPAlert.1.Destination

Header: content-type application/json

Auth: Basic or X token
```

8. Alert Configuration/SNMP Traps Configuration/Set Alert Destination 1 Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"SNMPAlert.1.Destination": "<new value you want to set>"}}
```

9. Alert Configuration/SNMP Traps Configuration/Test SNMP Trap

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DelliDRACCardService/Actions/DelliDRACCardService.SendTestSNMPTrap

Header: content-type application/json

Auth: Basic or X token

Body: {"InstanceID":"iDRAC.Embedded.1#SNMPAlert.<alert destination id>#Destination"}

Body Example: {"InstanceID":"iDRAC.Embedded.1#SNMPAlert.2#Destination"}
```

10. Alert Configuration/SNMP Settings/Get Community String

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes/SNMP.1.AgentCommunity

Header: content-type application/json

Auth: Basic or X token
```

11. Alert Configuration/SNMP Settings/Set Community String

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"SNMP.1.AgentCommunity": "<new value you want to set>"}}
```

12. Alert Configuration/SNMP Settings/Get SNMP Alert Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMP.1.AlertPort

Header: content-type application/json

Auth: Basic or X token
```

13. Alert Configuration/SNMP Settings/Set SNMP Alert Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMP.1.AlertPort": <pass in integer value from 1 to
65535>}}
```

14. Alert Configuration/SNMP Settings/Get SNMP Trap Format

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMP.1.TrapFormat

Header: content-type application/json

Auth: Basic or X token
```

15. Alert Configuration/SNMP Settings/Set SNMP Trap Format

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMP.1.TrapFormat": "<pass in SNMPv1, SNMPv2 or SNMPv3>"}}
```

16. Alert Configuration/SMTP(email) Configuration/Get email Alert 1 State

NOTE: Get and set examples are for destination 1. If you want to configure a different destination, change the index number.

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/EmailAlert.1.Enable

Header: content-type application/json

Auth: Basic or X token
```

17. Alert Configuration/SMTP(email) Configuration/Set email Alert 1 State

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"EmailAlert.1.Enable": "<pass in Enabled or Disabled>"}}
```

18. Alert Configuration/SMTP(email) Configuration/Get email Alert 1 Destination Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/EmailAlert.1.Address

Header: content-type application/json

Auth: Basic or X token
```

19. Alert Configuration/SMTP(email) Configuration/Set email Alert 1 Destination Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"EmailAlert.1.Address": "<new value you want to set>"}}
```

20. Alert Configuration/SMTP(email) Configuration/Get email Alert 1 Custom Message

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/EmailAlert.1.CustomMsg

Header: content-type application/json

Auth: Basic or X token
```

21. Alert Configuration/SMTP(email) Configuration/Set email Alert 1 Custom Message

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"EmailAlert.1.CustomMsg": "<new value you want to set>"}}
```

22. Alert Configuration/SMTP(email) Configuration/Send Test email

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DelliDRACCardService/Actions/Delli
DRACCardService.SendTestEmailAlert

Header: content-type application/json

Auth: Basic or X token

Body: {"InstanceID":"iDRAC.Embedded.1#EmailAlert.<destination id>#Address"}

Body Example: {"InstanceID":"iDRAC.Embedded.1#EmailAlert.1#Address"}
```

23. Alert Configuration/SMTP(email) Server Settings/Get SMTP Server IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPServerIPAddress

Header: content-type application/json

Auth: Basic or X token
```

24. Alert Configuration/SMTP(email) Server Settings/Set SMTP Server IP Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPServerIPAddress": "<new value you want
to set>"}}
```

25. Alert Configuration/SMTP(email) Server Settings/Get SMTP Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPPort

Header: content-type application/json

Auth: Basic or X token
```

26. Alert Configuration/SMTP(email) Server Settings/Set SMTP Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPPort": <pass in integer value 1 to
65535>}}
```

27. Alert Configuration/SMTP(email) Server Settings/Get SMTP Authentication

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPAuthentication

Header: content-type application/json

Auth: Basic or X token
```

28. Alert Configuration/SMTP(email) Server Settings/Set SMTP Authentication

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPAuthentication": "<pass in Enabled or
Disabled>"}}
```

29. Alert Configuration/SMTP(email) Server Settings/Get SMTP Username

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPUserName

Header: content-type application/json

Auth: Basic or X token
```

30. Alert Configuration/SMTP(email) Server Settings/Set SMTP Username

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPUserName": "<new value you want to
set>"}}
```

31. Alert Configuration/SMTP(email) Server Settings/Get SMTP Username Password

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPPassword

Header: content-type application/json

Auth: Basic or X token
```

32. Alert Configuration/SMTP(email) Server Settings/Set SMTP Username Password

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPPassword": "<new value you want to
set>"}}
```

33. Alert Configuration/SMTP(email) Server Settings/Get SMTP Connection Encryption

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.ConnectionEncryption

Header: content-type application/json

Auth: Basic or X token
```

34. Alert Configuration/SMTP(email) Server Settings/Set SMTP Connection Encryption

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.ConnectionEncryption": "<pass in None,
SSL/TLS or STARTTLS>"}}
```

35. Alert Configuration/Remote Syslog/Settings/Basic Settings/Get Remote Syslog Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.SysLogEnable

Header: content-type application/json
```

```
Auth: Basic or X token
```

36. Alert Configuration/Remote Syslog/Settings/Basic Settings/Set Remote Syslog Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.SysLogEnable": "<pass in Enabled or Disabled>"}}
```

37. Alert Configuration/Remote Syslog/Settings/Basic Settings/Get Syslog Server 1

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.Server1

Header: content-type application/json

Auth: Basic or X token
```

38. Alert Configuration/Remote Syslog/Settings/Basic Settings/Set Syslog Server 1

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.Server1": "<new value you want to set>"}}
```

39. Alert Configuration/Remote Syslog/Settings/Basic Settings/Get Syslog Server 2

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.Server2

Header: content-type application/json

Auth: Basic or X token
```

40. Alert Configuration/Remote Syslog/Settings/Basic Settings/Set Syslog Server 2

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.Server2": "<new value you want to set>"}}
```

41. Alert Configuration/Remote Syslog/Settings/Basic Settings/Get Syslog Server 3

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.Server3

Header: content-type application/json
```

```
Auth: Basic or X token
```

42. Alert Configuration/Remote Syslog/Settings/Basic Settings/Set Syslog Server 3

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.Server3": "<new value you want to set>"}}
```

43. Alert Configuration/Remote Syslog/Settings/Basic Settings/Get Syslog Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.Port

Header: content-type application/json

Auth: Basic or X token
```

44. Alert Configuration/Remote Syslog/Settings/Basic Settings/Get Syslog Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.Port": <pass in integer value from 1 to 65535>}}
```

45. Alert Configuration/Remote Syslog/Settings/Secure Settings/Get Security Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.securesyslogenable

Header: content-type application/json

Auth: Basic or X token
```

46. Alert Configuration/Remote Syslog/Settings/Secure Settings/Set Security Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.securesyslogenable": "<pass in Enabled or
Disabled>"}}
```

47. Alert Configuration/Remote Syslog/Settings/Secure Settings/Get Secure Syslog Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.secureserver1
```

```
Header: content-type application/json

Auth: Basic or X token
```

48. Alert Configuration/Remote Syslog/Settings/Secure Settings/Set Secure Syslog Server

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.secureserver1": "<new value you want to set>"}}
```

49. Alert Configuration/Remote Syslog/Settings/Secure Settings/Get Secure Port

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.secureport

Header: content-type application/json

Auth: Basic or X token
```

50. Alert Configuration/Remote Syslog/Settings/Secure Settings/Set Secure Port

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1
```

```
Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.secureport": <pass in integer value from 1 to
65535>}}
```

51. Alert Configuration/Remote Syslog/Settings/Secure Settings/Get Authentication

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysLog.1.secureclientauth

Header: content-type application/json

Auth: Basic or X token
```

52. Alert Configuration/Remote Syslog/Settings/Secure Settings/Set Secure Authentication

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SysLog.1.secureclientauth": "<pass in Anonymous or
Certificate>"}}
```

53. Alert Configuration/Remote Syslog/SSL/TLS Certificate Signing Request/Generate CSR

```
Command: POST

URI: redfish/v1/CertificateService/Actions/CertificateService.GenerateCSR
```

```
Header: content-type application/json

Auth: Basic or X token

Body:
{"CertificateCollection":{"@odata.id":"/redfish/v1/Managers/iDRAC.Embedded.1/Net
workProtocol/HTTPS/Certificates"}, "City": "<your city>","CommonName": "<your
common name>", "Country": "<your country>", "Organization": "<your org>",
"OrganizationalUnit": "<your org unit>","State": "<your state>", "email": "<your
email>"}
```

NOTE: In the JSON output, property CSRString value contains the CSR contents.

54. Alert Configuration/Remote Syslog/SSL/TLS Certificate Signing Request/Upload Signed Cert

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRAC
CardService.ImportSSLCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateType":"<pass in cert type, either CA, CSC,
ClientTrustCertificate, CustomCertificate, or
Server>","SSLCertificateFile":"<signed cert in base64 string format>"}
```

55. Alert Configuration/Test Event/Submit Test Event

```
Command: POST

URI: redfish/v1/EventService/Actions/EventService.SubmitTestEvent

Header: content-type application/json

Auth: Basic or X token

Body: {"Destination": "<URI path>", "EventTypes": "<event type>", "Context":
"Root", "Protocol": "Redfish", "MessageId": "CPU0001"}

Body Example: {"Destination": "https://192.168.0.130", "EventTypes": "Alert",
"Context": "Root", "Protocol": "Redfish", "MessageId": "CPU0001"}
```

NOTE: To get message IDs, run GET on URI “redfish/v1/Registries/Messages/EEMIRegistry”.

56. Alert Configuration/Get Critical Severity Alert Recurrence Frequency

```
Command: GET

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?
$select=Attributes/ThermalConfig.1.CriticalEventGenerationInterval

Header: content-type application/json

Auth: Basic or X token
```

57. Alert Configuration/Get Warning Severity Alert Recurrence Frequency

```
Command: GET

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?
$select=Attributes/ThermalConfig.1.EventGenerationInterval

Header: content-type application/json

Auth: Basic or X token
```

58. Alert Configuration/Set Critical Severity Alert Recurrence Frequency

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalConfig.1.CriticalEventGenerationInterval": "<pass
in integer value>"}}
```

59. Alert Configuration/Set Warning Severity Alert Recurrence Frequency

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalConfig.1.EventGenerationInterval": "<pass in
integer value>"}}
```

19 Section 19: Configuration/System Settings/Redfish Eventing

1. Redfish Event Settings/Get Maximum Number of Retries

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RedfishEventing.1.DeliveryRetryAttempts

Header: content-type application/json

Auth: Basic or X token
```

2. Redfish Event Settings/Set Maximum Number of Retries

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RedfishEventing.1.DeliveryRetryAttempts": <pass in
integer value from 0 to 5>}}
```

3. Redfish Event Settings/Get Retry Interval

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RedfishEventing.1.DeliveryRetryIntervalInSeconds

Header: content-type application/json

Auth: Basic or X token
```

4. Redfish Event Settings/Set Retry Interval

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RedfishEventing.1.DeliveryRetryIntervalInSeconds": <pass
in integer value from 5 to 60>}}
```

5. Redfish Event Settings/Get Ignore Certificate Errors

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RedfishEventing.1.IgnoreCertificateErrors

Header: content-type application/json

Auth: Basic or X token
```

6. Redfish Event Settings/Set Ignore Certificate Errors

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RedfishEventing.1.IgnoreCertificateErrors": "<pass in
value Yes or No>"}}
```

20 Section 20: Configuration/System Settings/Telemetry Configuration

NOTE: Telemetry configuration support requires the iDRAC Datacenter license.

1. Telemetry Streaming/Get Telemetry Data Stream

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Telemetry.1.EnableTelemetry

Header: content-type application/json

Auth: Basic or X token
```

2. Telemetry Streaming/Set Telemetry Data Stream

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Telemetry.1.EnableTelemetry": "<pass in Enabled or
Disabled>"}}
```

3. Telemetry Streaming/Get Rsyslog Server 1

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Telemetry.1.RSyslogServer1

Header: content-type application/json
```

```
Auth: Basic or X token
```

4. Telemetry Streaming/Set Rsyslog Server 1

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Telemetry.1.RSyslogServer1": "<new value you want to
set>"}}
```

5. Telemetry Streaming/Get Rsyslog Server 1 Port

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Telemetry.1.RSyslogServer1Port

Header: content-type application/json

Auth: Basic or X token
```

6. Telemetry Streaming/Set Rsyslog Server 1 Port

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Telemetry.1.RSyslogServer1Port": <pass in integer value
from 0 to 65535>}}
```

7. Telemetry Streaming/Get Rsyslog Server 2

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Telemetry.1.RSyslogServer2

Header: content-type application/json

Auth: Basic or X token
```

8. Telemetry Streaming/Set Rsyslog Server 2

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Telemetry.1.RSyslogServer2": "<new value you want to
set>"}}
```

9. Telemetry Streaming/Get Rsyslog Server 2 Port

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Telemetry.1.RSyslogServer2Port

Header: content-type application/json

Auth: Basic or X token
```

10. Telemetry Streaming/Set Rsyslog Server 2 Port

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Telemetry.1.RSyslogServer2Port": <pass in integer value
from 0 to 65535>}}
```

11. Telemetry Streaming/Get Telemetry Subscription 1

NOTE: Telemetry supports subscriptions 1 to 8. To get and set other subscriptions, change the last digit in the attribute name to Telemetry.1.TelemetrySubscription2, Telemetry.1.TelemetrySubscription3, so forth

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Telemetry.1.TelemetrySubscription1

Header: content-type application/json

Auth: Basic or X token
```

12. Telemetry Streaming/Set Telemetry Subscription 1

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Telemetry.1.TelemetrySubscription1": "<new value you want
to set>"}}
```

13. Metric Report Definition/AggregationMetrics/Get Enable State

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/TelemetryAggregationMetrics.1.EnableTelemetry

Header: content-type application/json

Auth: Basic or X token
```

14. Metric Report Definition/AggregationMetrics/Set Metric Report and Trigger (if supported)

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"TelemetryAggregationMetrics.1.EnableTelemetry":
"Enabled",

"TelemetryAggregationMetrics.1.ReportInterval":<pass in integer value>,

"TelemetryAggregationMetrics.1.RsyslogTarget":"<pass in TRUE or FALSE>",

"TelemetryAggregationMetrics.1.ReportTriggers":”<trigger type>"}}

Body Example:

{"Attributes": {"TelemetryAggregationMetrics.1.EnableTelemetry": "Enabled",

"TelemetryAggregationMetrics.1.ReportInterval":0,

"TelemetryAggregationMetrics.1.RsyslogTarget":"TRUE",

"TelemetryAggregationMetrics.1.ReportTriggers":"TMPCpuWarnTrigger"}}
```

NOTE: GET and SET commands here can be used for any metric report. Replace the first part of the attribute name with the metric report that you want to configure. (Example: Change “AggregationMetrics.1.EnableTelemetry” to “TelemetryMemorySensor.1.EnableTelemetry”. Here is the list of supported metric reports:

- TelemetryAggregationMetrics.1
- TelemetryCPUMemMetrics.1
- TelemetryCPURegisters.1
- TelemetryCPUSensor.1
- TelemetryFCPortStatistics.1
- TelemetryFCSensor.1
- TelemetryGPUMetrics.1
- TelemetryFanSensor.1
- TelemetryGPUMetrics.1
- TelemetryGPUStatistics.1
- TelemetryMemorySensor.1
- TelemetryNICSensor.1
- TelemetryNICStatistics.1
- TelemetryNVMeSMARTData.1
- TelemetryPSUMetrics.1
- TelemetryPowerMetrics.1
- TelemetryPowerStatistics.1
- TelemetrySensor.1
- TelemetrySerialLog.1
- TelemetryStorageDiskSMARTData.1
- TelemetryStorageSensor.1
- TelemetrySystemUsage.1
- TelemetryThermalMetrics.1
- TelemetryThermalSensor.1

NOTE: Before configuring Telemetry metric reports, recommended running GET on URI "redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=Attributes" for your iDRAC version, confirm supported metric reports.

15. Metric Report Definition/AggregationMetrics/Get Metric Report Trigger

NOTE: Some metric reports do not support configuring triggers. An empty attribute value in the GET response means that triggers are not supported.

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/TelemetryAggregationMetrics.1.ReportTriggers

Header: content-type application/json

Auth: Basic or X token
```

16. Metric Report Definition/AggregationMetrics/Set Metric Report Trigger

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"TelemetryAggregationMetrics.1.ReportTriggers": "<new
value you want to set>"}}
```

21 Section 21: Configuration/System Settings/Hardware Settings

1. Cooling Configuration/Get Thermal Profile Optimization

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.ThermalProfile

Header: content-type application/json

Auth: Basic or X token
```

2. Cooling Configuration/Set Thermal Profile Optimization

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.ThermalProfile": "<pass in Default
Thermal Profile Settings, Maximum Performance, Minimum Power or Sound Cap>"}}
```

3. Cooling Configuration/Get Fan Speed Offset

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.FanSpeedOffset

Header: content-type application/json

Auth: Basic or X token
```

4. Cooling Configuration/Set Fan Speed Offset

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.FanSpeedOffset": "<pass in Low, High,
Medium, Max or Off>"}}
```

5. Cooling Configuration/Get System Inlet Temperature Current Reading

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.SystemInletTemperature

Header: content-type application/json

Auth: Basic or X token
```

6. Cooling Configuration/Get System Exhaust Temperature Current Reading

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.SystemExhaustTemperature

Header: content-type application/json

Auth: Basic or X token
```

7. Cooling Configuration/Get Maximum PCIe Inlet Temperature Limit

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.MaximumPCIeInletTemperatureLimit

Header: content-type application/json

Auth: Basic or X token
```

8. Cooling Configuration/Set Maximum PCIe Inlet Temperature Limit

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.MaximumPCIeInletTemperatureLimit":
"<pass in 45 or 55 as a string value>"}}
```

9. Cooling Configuration/Get Maximum Exhaust Temperature Limit Setting

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.SetMaximumExhaustTemperatureLimit

Header: content-type application/json

Auth: Basic or X token
```

10. Cooling Configuration/Set Maximum Exhaust Temperature Limit Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.SetMaximumExhaustTemperatureLimit":
"<pass in Enabled or Disabled>"}}
```

11. Cooling Configuration/Get Maximum Exhaust Temperature Limit Current Range Value

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.AirExhaustTemp

Header: content-type application/json

Auth: Basic or X token
```

12. Cooling Configuration/Set Maximum Exhaust Temperature Limit Current Range Value

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.AirExhaustTemp": "<pass in 40, 45, 50,
55, 60, 65 or 70 as a string value>"}}
```

13. Cooling Configuration/Get Air Temperature Rise Limit Setting

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.SetAirTemperatureRiseLimit

Header: content-type application/json

Auth: Basic or X token
```

14. Cooling Configuration/Set Air Temperature Rise Limit Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.SetAirTemperatureRiseLimit": "<pass in
Enabled or Disabled>"}}
```

15. Cooling Configuration/Get Air Temperature Rise Limit Range Value

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/ThermalSettings.1.AirTemperatureRiseLimit

Header: content-type application/json

Auth: Basic or X token
```

16. Cooling Configuration/Set Air Temperature Rise Limit Range Value

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.AirTemperatureRiseLimit": "<pass in 15,
20, 25, 30, 35, 40, 45, 50 or NO LIMIT as a string value>"}}
```

17. Cooling Configuration/Get Thresholds Minimum Fan Speed in PWM (% of Max)

```
Command: GET

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?
$select=ThermalSettings.1.MinimumFanSpeed

Header: content-type application/json

Auth: Basic or X token
```

NOTE: The value 255 represents to use “Default” minimum fan speed and any other value represents “Custom” fan speed value in percentage.

18. Cooling Configuration/Set Thresholds Minimum Fan Speed in PWM (% of Max)

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ThermalSettings.1.MinimumFanSpeed": "<new value you want
to set that is within the range>"}}
```

NOTE: Pass 255 value to use “Default” minimum fan speed. The range is dynamic and varies depending on the system. Run GET on URI “/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$select=ThermalSettings.1.MFSMinimumLimit” to get the minimum value of the range and GET on URI “/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$select=ThermalSettings.1.MFSMaximumLimit” to get the maximum value of the range.

19. Front Panel Configuration/LCD Settings/Get Home Message

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/LCD.1.Configuration

Header: content-type application/json

Auth: Basic or X token
```

20. Front Panel Configuration/LCD Settings/Set Home Message

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"LCD.1.Configuration": "<new value you want to set>"}}
```

21. Front Panel Configuration/LCD Settings/Get Current Display Value

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/LCD.1.CurrentDisplay

Header: content-type application/json

Auth: Basic or X token
```

22. Front Panel Configuration/LCD Settings/Get Virtual Console Indication

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/LCD.1.vConsoleIndication

Header: content-type application/json

Auth: Basic or X token
```

23. Front Panel Configuration/LCD Settings/Set Virtual Console Indication

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"LCD.1.vConsoleIndication": "<pass in Enabled or
Disabled>"}}
```

24. Front Panel Configuration/System ID LED Settings/Get System LED Status

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1?$select=IndicatorLED

Header: content-type application/json

Auth: Basic or X auth
```

25. Front Panel Configuration/System ID LED Settings/Set System LED Status

```
Command: PATCH

URI: redfish/v1/Chassis/System.Embedded.1

Header: content-type application/json

Auth: Basic or X auth

Body: {"IndicatorLED":"Blinking"}
```

NOTE: Supported values are Blinking and Lit.

26. iDRAC Quick Sync/Get Presence

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/QuickSync.1.Presence

Header: content-type application/json

Auth: Basic or X auth
```

27. iDRAC Quick Sync/Get Access

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/QuickSync.1.Access

Header: content-type application/json

Auth: Basic or X auth
```

28. iDRAC Quick Sync/Set Access

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"QuickSync.1.Access": "<pass in Disabled, Read-only or
Read-write>"}}
```

29. iDRAC Quick Sync/Get Timeout

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/QuickSync.1.InactivityTimerEnable

Header: content-type application/json

Auth: Basic or X auth
```

30. iDRAC Quick Sync/Set Timeout

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1
```

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"QuickSync.1.InactivityTimerEnable": "<pass in Enabled or Disabled>"}}

31. iDRAC Quick Sync/Get Timeout Limit

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$select=Attributes/QuickSync.1.InactivityTimeout

Header: content-type application/json

Auth: Basic or X auth

32. iDRAC Quick Sync/Set Timeout Limit

Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"QuickSync.1.InactivityTimeout": <pass in integer value from 120 to 3600>}}

33. iDRAC Quick Sync/Get Read Authentication

Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$select=Attributes/QuickSync.1.ReadAuthentication

Header: content-type application/json

Auth: Basic or X auth

34. iDRAC Quick Sync/Set Read Authentication

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"QuickSync.1.ReadAuthentication": "<pass in Enabled or
Disabled>"}}
```

35. iDRAC Quick Sync/Get Wi-Fi

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/QuickSync.1.WifiEnable

Header: content-type application/json

Auth: Basic or X auth
```

36. iDRAC Quick Sync/Set Wi-Fi

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"QuickSync.1.WifiEnable": "<pass in Enabled or
Disabled>"}}
```

37. First Boot Device/Get First Boot Device

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1?$select=Boot/BootSourceOverrideTarget

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Value of None means that no onetime boot device is set.

38. First Boot Device/Get Supported Values for Set First Boot Device

```
Command: GET

URI:
redfish/v1/Systems/System.Embedded.1?$select=Boot/BootSourceOverrideTarget@Redfish.AllowableValues

Header: content-type application/json

Auth: Basic or X auth
```

39. First Boot Device/Set First Boot Device

```
Command: PATCH

URI: redfish/v1/Systems/System.Embedded.1

Header: content-type application/json

Auth: Basic or X auth

Body: {"Boot":{"BootSourceOverrideTarget":"<your onetime boot device value>"}}
```

NOTE: You must now reboot the server to perform onetime boot to the device you set.

40. Front Ports/Get Front USB Port Setting

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/USBFront.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

41. Front Ports/Set Front USB Port Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"USBFront.1.Enable": "<pass in Enabled or Disabled>"}}
```

42. I/O Identity Optimization/Get I/O Identity Optimization Setting

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IOIDOpt.1.IOIDOptEnable

Header: content-type application/json

Auth: Basic or X auth
```

43. I/O Identity Optimization/Set I/O Identity Optimization Setting

```
Command: PATCH
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IOIDOpt.1.IOIDOptEnable": "<pass in Enabled or
Disabled>"}}
```

44. I/O Identity Optimization/Persistent Policy/Get Virtual Address Auxiliary Powered Devices

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IOIDOpt.1.VirtualAddressPersistencePolicyAuxPwrd

Header: content-type application/json

Auth: Basic or X auth
```

45. I/O Identity Optimization/Set Virtual Address Auxiliary Powered Devices

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IOIDOpt.1.VirtualAddressPersistencePolicyAuxPwrd": "<new
value(s) you want to set>"}}
```

Note: Supported values are: WarmReset, ColdReset, ACPowerLoss. You can pass in one or multiple values as one string value. Example: “WarmReset, ColdReset”

46. I/O Identity Optimization/Persistent Policy/Get Virtual Address Non-Auxiliary Powered Devices

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IOIDOpt.1.VirtualAddressPersistencePolicyNonAuxPwrd

Header: content-type application/json

Auth: Basic or X auth
```

47. I/O Identity Optimization/Set Virtual Address Non-Auxiliary Powered Devices

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IOIDOpt.1.VirtualAddressPersistencePolicyNonAuxPwrd":
"<new value(s) you want to set>"}}
```

Note: Supported values are: WarmReset, ColdReset, ACPowerLoss. You can pass in one or multiple values as one string value. Example: “WarmReset, ColdReset”

48. I/O Identity Optimization/Persistent Policy/Get Initiator

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IOIDOpt.1.InitiatorPersistencePolicy

Header: content-type application/json

Auth: Basic or X auth
```

49. I/O Identity Optimization/Set Initiator

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IOIDOpt.1.InitiatorPersistencePolicy": "<new value(s) you
want to set>"}}
```

Note: Supported values are: WarmReset, ColdReset, ACPowerLoss. You can pass in one or multiple values as one string value. Example: “WarmReset, ColdReset”

50. I/O Identity Optimization/Persistent Policy/Get Storage Target

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IOIDOpt.1.StorageTargetPersistencePolicy

Header: content-type application/json

Auth: Basic or X auth
```

51. I/O Identity Optimization/Set Storage Target

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IOIDOpt.1.StorageTargetPersistencePolicy": "<new value(s)
you want to set>"}}
```

Note: Supported values are: WarmReset, ColdReset, ACPowerLoss. You can pass in one or multiple values as one string value. Example: “WarmReset, ColdReset”

52. SSD Wear Thresholds/Get Remaining Read Write Endurance Alert Thresholds

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/Storage.1.RemainingRatedWriteEnduranceAlertThreshold

Header: content-type application/json

Auth: Basic or X auth
```

53. SSD Wear Thresholds/Set Remaining Read Write Endurance Alert Thresholds

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Storage.1.RemainingRatedWriteEnduranceAlertThreshold":
"<new value you want to set>"}}
```

54. SSD Wear Thresholds/Get Available Spare Alert Thresholds

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1?$
select=Attributes/Storage.1.AvailableSpareAlertThreshold

Header: content-type application/json
```

```
Auth: Basic or X auth
```

55. SSD Wear Thresholds/Set Available Spare Alert Thresholds

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/System.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Storage.1.AvailableSpareAlertThreshold": <pass in integer
value from 1 to 99>}}
```

22 Section 22: Configuration/BIOS Settings

1. Get BIOS Attributes

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Bios?$select=Attributes

Header: content-type application/json

Auth: Basic or X auth
```

2. Set BIOS Attributes

```
Command: PATCH

URI: redfish/v1/Systems/System.Embedded.1/Bios/Settings

Header: content-type application/json

Auth: Basic or X auth

Body: {"@Redfish.SettingsApplyTime": {"ApplyTime": "OnReset"}, "Attributes":
{"<attribute name>": "<attribute value>"}}

Body example: {{"@Redfish.SettingsApplyTime": {"ApplyTime": "OnReset"},
"Attributes": {"EmbSata": "RaidMode", "NvmeMode": "Raid"}}
```

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Reboot the server to run the config job.

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "GracefulRestart"}
```

NOTE: Once you reboot the server, run GET command again on the same job ID URI to monitor the status until it is marked as completed.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

3. Get BIOS Boot Order

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/BootOptions?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: For each boot order device entry, property ID value is used for changing the boot order.

4. Change BIOS Boot Order

```
Command: PATCH

URI: redfish/v1/Systems/System.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X auth

Body: {"Boot": {"BootOrder": ["<boot order device id(s)"]}}

Body example: {"Boot": {"BootOrder": ["Boot0000", "Boot0008", "Boot0009"]}}
```

NOTE: For boot order device IDs you can pass in one, multiple or all boot order devices. If you only pass in one boot order device Id, that device is set as first device. The rest of the boot order is shifted down.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Reboot the server to run the config job.

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "GracefulRestart"}
```

NOTE: Once you reboot the server, run GET command again on the same job ID URI to monitor the status until it is marked as completed.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

Success status code returned: 200

5. Get Current BIOS Boot Order Enabled/Disabled State

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/BootOptions?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: For each boot order device entry, property BootOptionEnabled value is used to determine if boot entry is enabled (true) or disabled (false).

6. Disable Boot Order Device

```
Command: PATCH

URI: redfish/v1/Systems/System.Embedded.1/BootOptions/{boot order device ID}

URI example: /redfish/v1/Systems/System.Embedded.1/BootOptions/Boot0000

Header: content-type application/json

Auth: Basic or X auth

Body: {"BootOptionEnabled":False}
```

NOTE: Boolean value for property BootOptionEnabled. If you want to enable a boot order device, pass in a value of True.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json
```

```
Auth: Basic or X auth
```

NOTE: Reboot the server to run the config job.

```
Command: POST

URI: redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "GracefulRestart"}
```

NOTE: Once you reboot the server, run GET command again on the same job ID URI to monitor the status until it is marked as completed.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

23 Section 23: Configuration/Server Configuration Profile

1. Export Server Configuration Profile Locally

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ExportSystemCon
figuration

Header: content-type application/json

Auth: Basic or X auth

Body: {"ExportFormat": "<pass in XML or JSON>", "ShareParameters": {"Target":
"ALL"}}
```

NOTE: For all supported parameters for this action, see schema “redfish/v1/Schemas/OemManager_v1.xml”.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Once the job ID is marked completed, the SCP file content is in the JSON output.

2. Export Server Configuration Profile Network Share

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ExportSystemCon
figuration

Header: content-type application/json
```

```
Auth: Basic or X auth

Body: {"ExportFormat": "XML", "ShareParameters": {"Target": "ALL", "IPAddress":
"<network share ip>", "ShareType": "<pass in NFS, CIFS, HTTP, or HTTPS",
"ShareName": "<your share name>", "FileName": "<name your SCP file>"}}

Body Example: {"ExportFormat": "XML", "ShareParameters": {"Target": "ALL",
"IPAddress": "192.168.0.130", "ShareType": "NFS", "ShareName": "/nfs",
"FileName": "R640_SCP_file.xml"}}
```

NOTE: For all supported parameters for this action, see schema “redfish/v1/Schemas/OemManager_v1.xml”.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

3. Preview Import Server Configuration Profile Locally

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ImportSystemCon
figurationPreview

Header: content-type application/json

Auth: Basic or X auth

Body: {"ImportBuffer": "<pass in the complete SCP file as a string value",
"ShareParameters": {"Target": "ALL"}}

Body Example: {{"ImportBuffer": "<SystemConfiguration Model=\"WWT_TEST\"
ServiceTag=\"M536C3S\" TimeStamp=\"Thu Oct 13 22:06:18 2022\"><!--Export type is
Normal,XML,Selective--><!--Exported configuration may contain commented
attributes. Attributes may be commented due to dependency, destructive nature,
preserving server identity or for security reasons.--><Component
FQDD=\"LifecycleController.Embedded.1\"><Attribute
Name=\"LCAttributes.1#CollectSystemInventoryOnRestart\">Enabled</Attribute><Attr
ibute Name=\"LCAttributes.1#PartConfigurationUpdate\">Apply
```

```
Always</Attribute><Attribute Name=\"LCAttributes.1#PartFirmwareUpdate\">Match
firmware of replaced part</Attribute><!-- <Attribute
Name=\"LCAttributes.1#LifecycleControllerState\">Enabled</Attribute>--
><Attribute Name=\"LCAttributes.1#IPChangeNotifyPS\">Off</Attribute><Attribute
Name=\"LCAttributes.1#VirtualAddressManagementApplication\"></Attribute><Attribu
te Name=\"LCAttributes.1#ProvisioningServer\"></Attribute><Attribute
Name=\"LCAttributes.1#BIOSRTDRequested\">False</Attribute><Attribute
Name=\"LCAttributes.1#AutoUpdate\">Disabled</Attribute><Attribute
Name=\"LCAttributes.1#IPAddress\"></Attribute><Attribute
Name=\"LCAttributes.1#UserProxyServer\"></Attribute><Attribute
Name=\"LCAttributes.1#UserProxyPort\">80</Attribute><Attribute
Name=\"LCAttributes.1#UserProxyUserName\"></Attribute><!-- <Attribute
Name=\"LCAttributes.1#UserProxyPassword\">******</Attribute>--><Attribute
Name=\"LCAttributes.1#UserProxyType\">HTTP</Attribute><Attribute
Name=\"LCAttributes.1#IgnoreCertWarning\">On</Attribute><!-- <Attribute
Name=\"OSD.1#SupportedOSList\"></Attribute>--><Attribute
Name=\"OSD.1#OSName\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareIP\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareName\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareUsername\"></Attribute><Attribute
Name=\"OSD.1#OSMediaSharePassword\">******</Attribute><Attribute
Name=\"OSD.1#OSMediaShareDomainName\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareType\"></Attribute><Attribute
Name=\"OSD.1#OSMediaName\"></Attribute><Attribute
Name=\"OSD.1#AnswerFileName\"></Attribute><Attribute
Name=\"OSD.1#ExposeDuration\"></Attribute><Attribute
Name=\"OSD.1#OSMediaHashType\"></Attribute><Attribute
Name=\"OSD.1#OSMediaHashValue\"></Attribute></Component></SystemConfiguration>",
"ShareParameters": {"Target": "ALL"}}
```

NOTE: For all supported parameters for this action, see schema “redfish/v1/Schemas/OemManager_v1.xml”.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

4. Preview Import Server Configuration Profile Network Share

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ImportSystemConfigurationPreview

Header: content-type application/json

Auth: Basic or X auth

Body: {"ExportFormat": "<pass in XML or JSON>", "ShareParameters": {"Target": "ALL", "IPAddress": "<share ip>", "ShareType": "<share type>", "ShareName": "<share name>", "FileName": "<your SCP file>"}}

Body Example: {"ExportFormat": "XML", "ShareParameters": {"Target": "ALL", "IPAddress": "192.168.0.130", "ShareType": "HTTP", "ShareName": "http_share", "FileName": "R650_SCP_file.xml"}}
```

NOTE: For all supported parameters for this action, see schema “redfish/v1/Schemas/OemManager_v1.xml”.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

5. Import Server Configuration Profile Locally

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ImportSystemConfiguration

Header: content-type application/json

Auth: Basic or X auth

Body: {"ImportBuffer": "<pass in the complete SCP file as a string value", "ShareParameters": {"Target": "ALL"}}
```

```
Body Example: {{"ImportBuffer": "<SystemConfiguration Model=\"WWT_TEST\"
ServiceTag=\"M536C3S\" TimeStamp=\"Thu Oct 13 22:06:18 2022\"><!--Export type is
Normal,XML,Selective--><!--Exported configuration may contain commented
attributes. Attributes may be commented due to dependency, destructive nature,
preserving server identity or for security reasons.--><Component
FQDD=\"LifecycleController.Embedded.1\"><Attribute
Name=\"LCAttributes.1#CollectSystemInventoryOnRestart\">Enabled</Attribute><Attr
ibute Name=\"LCAttributes.1#PartConfigurationUpdate\">Apply
Always</Attribute><Attribute Name=\"LCAttributes.1#PartFirmwareUpdate\">Match
firmware of replaced part</Attribute><!-- <Attribute
Name=\"LCAttributes.1#LifecycleControllerState\">Enabled</Attribute>--
><Attribute Name=\"LCAttributes.1#IPChangeNotifyPS\">Off</Attribute><Attribute
Name=\"LCAttributes.1#VirtualAddressManagementApplication\"></Attribute><Attribu
te Name=\"LCAttributes.1#ProvisioningServer\"></Attribute><Attribute
Name=\"LCAttributes.1#BIOSRTDRequested\">False</Attribute><Attribute
Name=\"LCAttributes.1#AutoUpdate\">Disabled</Attribute><Attribute
Name=\"LCAttributes.1#IPAddress\"></Attribute><Attribute
Name=\"LCAttributes.1#UserProxyServer\"></Attribute><Attribute
Name=\"LCAttributes.1#UserProxyPort\">80</Attribute><Attribute
Name=\"LCAttributes.1#UserProxyUserName\"></Attribute><!-- <Attribute
Name=\"LCAttributes.1#UserProxyPassword\">******</Attribute>--><Attribute
Name=\"LCAttributes.1#UserProxyType\">HTTP</Attribute><Attribute
Name=\"LCAttributes.1#IgnoreCertWarning\">On</Attribute><!-- <Attribute
Name=\"OSD.1#SupportedOSList\"></Attribute>--><Attribute
Name=\"OSD.1#OSName\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareIP\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareName\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareUsername\"></Attribute><Attribute
Name=\"OSD.1#OSMediaSharePassword\">******</Attribute><Attribute
Name=\"OSD.1#OSMediaShareDomainName\"></Attribute><Attribute
Name=\"OSD.1#OSMediaShareType\"></Attribute><Attribute
Name=\"OSD.1#OSMediaName\"></Attribute><Attribute
Name=\"OSD.1#AnswerFileName\"></Attribute><Attribute
Name=\"OSD.1#ExposeDuration\"></Attribute><Attribute
Name=\"OSD.1#OSMediaHashType\"></Attribute><Attribute
Name=\"OSD.1#OSMediaHashValue\"></Attribute></Component></SystemConfiguration>",
"ShareParameters": {"Target": "ALL"}}
```

NOTE: For all supported parameters for this action, see schema “redfish/v1/Schemas/OemManager_v1.xml”.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json
```

```
Auth: Basic or X auth
```

6. Import Server Configuration Profile Network Share

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/EID_674_Manager.ImportSystemCon
figuration

Header: content-type application/json

Auth: Basic or X auth

Body: {"ExportFormat": "<pass in XML or JSON>", "ShareParameters": {"Target":
"ALL", "IPAddress": "<share ip>", "ShareType": "<share type>", "ShareName":
"<share name>", "FileName": "<your SCP file>"}}

Body Example: {"ExportFormat": "XML", "ShareParameters": {"Target": "ALL",
"IPAddress": "192.168.0.130", "ShareType": "HTTP", "ShareName": "http_share",
"FileName": "R650_SCP_file.xml"}}
```

NOTE: For all supported parameters for this action, see schema “redfish/v1/Schemas/OemManager_v1.xml”.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

7. Upload Custom Defaults Server Configuration Profile Locally

```
Command: POST

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/DellManager.SetCustomDefaults

Header: content-type application/json

Auth: Basic or X auth

Body: {"CustomDefaults": "<pass in the complete SCP file as a string value"}
```

24 Section 24: Maintenance/Lifecycle Log

1. Export Lifecycle Log Locally

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portLCLog

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType": "Local", "FileName": "lclog.xml"}
```

NOTE: LC log URI file location is in the headers location. URI: redfish/v1/Dell/lclog.xml

2. Export Lifecycle Log Network Share

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portLCLog

Header: content-type application/json

Auth: Basic or X auth

Body: {"IPAddress": "<share ip>", "ShareType": "<share type>", "ShareName":
"<share name>", "FileName": "<unique file name>"}

Body Example: {"IPAddress": "192.168.0.130", "ShareType": "HTTP", "ShareName":
"http_share", "FileName": "R640_lclog.xml"}
```

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status to show Scheduled.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json
```

```
Auth: Basic or X auth
```

25 Section 25: Maintenance/Job Queue

1. Get current iDRAC job queue

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

2. Delete one job ID

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellJobService/Actions/DellJobService.
DeleteJobQueue

Header: content-type application/json

Auth: Basic or X auth

Body: {"JobID":"<job ID you want to delete>"}

Body Example: {"JobID":"JID_631039262553"}
```

NOTE: You cannot delete a job in running state, only downloaded, scheduled, completed, or failed.

3. Delete all job IDs in the job queue

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellJobService/Actions/DellJobService.
DeleteJobQueue

Header: content-type application/json

Auth: Basic or X auth
```

```
Body: {"JobID":"JID_CLEARALL"}
```

NOTE: This command fails if a job is in running state.

26 Section 26: Maintenance/System Update

1. Manual Update (One Device)

```
Command: POST

URI: redfish/v1/UpdateService/MultipartUpload

Header/File: {"UpdateParameters": (None, json.dumps({"Targets": [],
"@Redfish.OperationApplyTime": "Immediate", "Oem": {}}),
"application/json"),"UpdateFile": ("<name of firmware image>"), open("<complete
path of directory location and firmware image name>", "rb"), "application/octet-
stream")}

Header/File Example: {"UpdateParameters": (None, \"{"Targets": [],
"@Redfish.OperationApplyTime": "Immediate", "Oem": {}}\", "application/json"),
"UpdateFile": ("BIOS_DHRG5_WN64_1.7.5.EXE", <_io.BufferedReader
name="C:\\Users\\administrator\\Downloads\\BIOS_DHRG5_WN64_1.7.5.EXE">,
"application/octet-stream")}

Auth: Basic or X auth
```

NOTE: View script on GitHub for more details and coding examples.

https://github.com/dell/iDRAC-Redfish-Scripting/blob/master/Redfish%20Python/DeviceFirmwareMultipartUploadREDFISH.py

NOTE: Only Windows Dell Update Packages are supported for firmware updates using Redfish.

NOTE: If the server requires a reboot to apply the firmware update (Examples: BIOS or PERC), server automatically reboots.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status until it is marked as completed.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

2. Rollback (One Device)

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: In the JSON output Members, look for any URI with “Previous” string, which means that this device supports rollback.

Example: /redfish/v1/UpdateService/FirmwareInventory/Previous-25227-3.32.32.32__iDRAC.Embedded.1-1

```
Command: POST

URI: redfish/v1/UpdateService/Actions/UpdateService.SimpleUpdate

Header: content-type application/json

Auth: Basic or X auth

Body: {"ImageURI":"<previous device URI>"}

Body Example: {"ImageURI":"/redfish/v1/UpdateService/FirmwareInventory/Previous-
25227-3.32.32.32__iDRAC.Embedded.1-1"}
```

NOTE: Job ID URI is returned in the headers location. Run GET command on below URI to validate JobState marked as Scheduled or Completed. If marked Scheduled, a server reboot is required to run the job.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

3. Get Automatic Schedule Update Settings

```
Command: POST
```

URI:
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellSoftwareInstallationService/Actions/DellSoftwareInstallationService.GetUpdateSchedule

Header: content-type application/json

Auth: Basic or X auth

Body: {}

4. Enable/Disable Automatic Update

Command: PATCH

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/LifecycleController.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {" LCAttributes.1.AutoUpdate": "<pass in Enabled or Disabled>"}}

5. Set Automatic Schedule Update

Command: POST

URI:
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellSoftwareInstallationService/Actions/DellSoftwareInstallationService.SetUpdateSchedule

Header: content-type application/json

Auth: Basic or X auth

Body: {"ApplyReboot":"<pass value>” ,"IPAddress":"<pass IP Address>
","Repeat":<pass
value>,"DayofMonth":"*","WeekofMonth":"*","DayofWeek":"*","ShareType":"<pass value>","Time":"<pass value>","IgnoreCertWarning":"<pass value>"}

Note: Same properties to be passed in Request Body as present in GetUpdateSchedule API.

6. Clear Automatic Update Settings

```
Command: POST

URI:
/redfish/v1/Systems/System.Embedded.1/Oem/Dell/DellSoftwareInstallationService/A
ctions/DellSoftwareInstallationService.ClearUpdateSchedule

Header: content-type application/json

Auth: Basic or X auth

Body: {}
```

27 Section 27: Maintenance/System event log

1. Get System event log

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/LogServices/Sel/Entries

Header: content-type application/json

Auth: Basic or X auth
```

2. Clear System event log

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/LogServices/Sel/Actions/LogService.ClearLog

Header: content-type application/json

Auth: Basic or X auth

Body: {}
```

NOTE: You must pass in an empty JSON body.

28 Section 28: Maintenance/Troubleshooting

1. Video Capture/Export Boot Capture Videos

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portVideoLog

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType": "Local", "FileType": "BootCaptureVideo"}
```

NOTE: URI location of the videos is in the headers location. URI: /redfish/v1/Dell/bootlogs.zip

2. POST Code/Get POST Code

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SysInfo.1.POSTCode

Header: content-type application/json

Auth: Basic or X auth
```

3. Intrusion/Get Intrusion Status

```
Command: GET

URI: redfish/v1/Chassis/System.Embedded.1?$select=PhysicalSecurity

Header: content-type application/json

Auth: Basic or X auth
```

4. Last Crash Screen/Export Last Crash Screen

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.Ex
portVideoLog

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType": "Local", "FileType": "CrashCaptureVideo"}
```

NOTE: URI location of the videos is in the headers location. URI: /redfish/v1/Dell/crashlogs.zip

29 Section 29: Maintenance/Diagnostics

1. Reboot iDRAC

```
Command: POST

URI: redfish/v1/Managers/iDRAC.Embedded.1/Actions/Manager.Reset

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "GracefulRestart"}
```

2. Reset iDRAC to Default Settings

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Actions/Oem/DellManager.ResetToDefaults

Header: content-type application/json

Auth: Basic or X auth

Body: {"ResetType": "<pass in reset type"}
```

Supported values for ResetType:

- All (All configurations are set to default).
- ResetAllWithRootDefaults (All configurations including network are set to default. Exception root user password set to calvin)
- Default (All configurations are set to default except users and network settings).
- CustomDefaults(All configurations are set to customer default settings, and this option will be available only if a customer defaults settings have been uploaded using Sever Configuration Profile).

3. Serial Data Logs/Get Serial Data Connection Setting

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SerialCapture.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

4. Serial Data Logs/Enable Serial Data Connection Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SerialCapture.1.Enable": "Enabled"}}
```

5. Serial Data Logs/Export Logs

```
Command: POST

URI:
redfish/v1/Managers/iDRAC.Embedded.1/SerialInterfaces/Serial.1/Actions/Oem/DellS
erialInterface.SerialDataExport

Header: content-type application/json

Auth: Basic or X auth

Body: {}
```

NOTE: Pass in empty JSON body.

NOTE: Serial logs are in the JSON content.

6. Serial Data Logs/Clear Logs

```
Command: POST
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/SerialInterfaces/Serial.1/Actions/Oem/DellS
erialInterface.SerialDataClear

Header: content-type application/json

Auth: Basic or X auth

Body: {}
```

NOTE: Pass in empty JSON body.

7. Serial Data Logs/Disable Serial Data Connection Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SerialCapture.1.Enable": "Disabled"}}
```

30 Section 30: Maintenance/SupportAssist

1. Accept End User License Agreement (EULA)

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.SupportAssistAcceptEULA

Header: content-type application/json

Auth: Basic or X auth

Body: {}
```

NOTE: You must pass in empty JSON body.

2. Export SupportAssist Collection Locally

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.SupportAssistCollection

Header: content-type application/json

Auth: Basic or X auth

Body: {"ShareType": "Local", "DataSelectorArrayIn": ["<pass in data types for collection>"]}

Body Example: {"ShareType": "Local", "DataSelectorArrayIn": ["HWData", "OSAppData"]}
```

NOTE: See schema “redfish/v1/Schemas/DellLCService_v1.xml” for details about supported parameters/values for the action.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status until marked Completed.

NOTE: Once the job ID is marked completed, the headers location returns URI where the SA collection can be downloaded “/redfish/v1/Dell/sacollect.zip”.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

3. Export SupportAssist Collection Network Share

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellLCService/Actions/DellLCService.SupportAssistCollection

Header: content-type application/json

Auth: Basic or X auth

Body: {"IPAddress": "<network share ip>", "ShareType": "<pass in HTTP, HTTPS, NFS, or CIFS>", "ShareName": "<your share name>", "DataSelectorArrayIn": ["<pass in data types for collection>"]}

Body Example: {"IPAddress": "192.168.0.130", "ShareType": "HTTP", "ShareName": "http_share", "DataSelectorArrayIn": ["HWData"]}
```

NOTE: See schema “redfish/v1/Schemas/DellLCService_v1.xml” for details about supported parameters/values for the action.

NOTE: JOB ID URI is returned in JSON response headers location. Run GET command on this URI to monitor the job status until marked Completed.

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Jobs/{job ID}

Header: content-type application/json

Auth: Basic or X auth
```

31 Section 31: iDRAC Settings/Overview

1. iDRAC Details/Device Type

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Info.1.Type

Header: content-type application/json

Auth: Basic or X auth
```

2. iDRAC Details/Hardware Version

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Info.1.HWRev

Header: content-type application/json

Auth: Basic or X auth
```

3. iDRAC Details/Firmware Version

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Info.1.Version

Header: content-type application/json

Auth: Basic or X auth
```

4. iDRAC Details/Firmware Updated

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory/Installed-25227-<your iDRAC
version>__iDRAC.Embedded.1-
1?$select=Oem/Dell/DellSoftwareInventory/InstallationDate

URI Example: redfish/v1/UpdateService/FirmwareInventory/Installed-25227-
6.00.02.00__iDRAC.Embedded.1-
1?$select=Oem/Dell/DellSoftwareInventory/InstallationDate

Header: content-type application/json

Auth: Basic or X auth
```

5. iDRAC Details/RAC Time

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DellTimeService/Actions/DellTimeServic
e.ManageTime

Header: content-type application/json

Auth: Basic or X auth

Body: {"GetRequest":true}
```

6. iDRAC Details/Number of Possible Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.MaxNumberOfSessions

Header: content-type application/json

Auth: Basic or X auth
```

7. iDRAC Details/Number of Current Sessions

```
Command: GET

URI: redfish/v1/SessionService/Sessions?$select=Members@odata.count

Header: content-type application/json

Auth: Basic or X auth
```

8. iDRAC Details/IPMI Version

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Info.1.IPMIVersion

Header: content-type application/json

Auth: Basic or X auth
```

9. iDRAC Details/Get User Interface Title Bar Information

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.TitleBarOption

Header: content-type application/json

Auth: Basic or X auth
```

10. iDRAC Details/Set User Interface Title Bar Information

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1
```

```
Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.TitleBarOption": "<value you want to set>"}}
```

NOTE: Possible values are Auto, DNS RAC Name, IP Address, Service Tag, System Host Name, Custom.

11. iDRAC Service Module/Status

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.ServiceModuleState

Header: content-type application/json

Auth: Basic or X auth
```

12. Connection View/State

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SwitchConnectionView.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

13. Connection View/Switch Connection Details

```
Command: GET

URI:
redfish/v1/Dell/Systems/System.Embedded.1/NetworkPorts/DellSwitchConnectionColle
ction

Header: content-type application/json
```

```
Auth: Basic or X auth
```

14. Current Network Settings/iDRAC MAC Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentNIC.1.MACAddress

Header: content-type application/json

Auth: Basic or X auth
```

15. Current Network Settings/Active NIC Interface

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentNIC.1.ActiveNIC

Header: content-type application/json

Auth: Basic or X auth
```

16. Current Network Settings/DNS Domain Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentNIC.1.DNSDomainName

Header: content-type application/json

Auth: Basic or X auth
```

17. Current IPv4 Settings/IPv4 Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

18. Current IPv4 Settings/DHCP

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.DHCPEnable

Header: content-type application/json

Auth: Basic or X auth
```

19. Current IPv4 Settings/Current IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.Address

Header: content-type application/json

Auth: Basic or X auth
```

20. Current IPv4 Settings/Current Subnet Mask

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.Netmask

Header: content-type application/json

Auth: Basic or X auth
```

21. Current IPv4 Settings/Current Gateway

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.Gateway

Header: content-type application/json

Auth: Basic or X auth
```

22. Current IPv4 Settings/Use DHCP to Obtain DNS Server Addresses

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentNIC.1.DNSDomainFromDHCP

Header: content-type application/json

Auth: Basic or X auth
```

23. Current IPv4 Settings/Current Preferred DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.DNS1
```

```
Header: content-type application/json

Auth: Basic or X auth
```

24. Current IPv4 Settings/Current Alternate DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv4.1.DNS2

Header: content-type application/json

Auth: Basic or X auth
```

25. Current IPv6 Settings/IPv6 Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

26. Current IPv6 Settings/Autoconfiguration

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.AutoConfig

Header: content-type application/json

Auth: Basic or X auth
```

27. Current IPv6 Settings/Current IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.Address1

Header: content-type application/json

Auth: Basic or X auth
```

28. Current IPv6 Settings/Current IP Gateway

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.Gateway

Header: content-type application/json

Auth: Basic or X auth
```

29. Current IPv6 Settings/Link Local Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.LinkLocalAddress

Header: content-type application/json

Auth: Basic or X auth
```

30. Current IPv6 Settings/Use DHCP to Obtain DNS Server Addresses

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.DNSFromDHCP6

Header: content-type application/json

Auth: Basic or X auth
```

31. Current IPv6 Settings/Current Preferred DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.DNS1

Header: content-type application/json

Auth: Basic or X auth
```

32. Current IPv6 Settings/Current Alternate DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentIPv6.1.DNS2

Header: content-type application/json

Auth: Basic or X auth
```

32 Section 32: iDRAC Settings/Connectivity

1. Network/Network Settings/Get Enable NIC

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

2. Network/Network Settings/Set Enable NIC

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.Enable": "<pass in Enabled or Disabled>"}}
```

3. Network/Network Settings/Get NIC Selection

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.Selection

Header: content-type application/json

Auth: Basic or X auth
```

4. Network/Network Settings/Set NIC Selection

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.Selection": "<pass in Dedicated, LOM1, LOM2, LOM3,
or LOM4>"}}
```

5. Network/Network Settings/Get Failover Network

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.Failover

Header: content-type application/json

Auth: Basic or X auth
```

6. Network/Network Settings/Set Failover Network

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.Failover": "<pass in All, None, LOM2, LOM3, or
LOM4>"}}
```

7. Network/Network Settings/Get Auto Dedicated NIC

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.AutoDetect

Header: content-type application/json

Auth: Basic or X auth
```

8. Network/Network Settings/Set Auto Dedicated NIC

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.AutoDetect": "<pass in Disabled or Enabled>"}}
```

9. Network/Network Settings/Get Active NIC Interface

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/CurrentNIC.1.ActiveNIC

Header: content-type application/json

Auth: Basic or X auth
```

10. Network/Network Settings/Get Auto Negotiation

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.Autoneg

Header: content-type application/json

Auth: Basic or X auth
```

11. Network/Network Settings/Set Auto Negotiation

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.Autoneg": "<pass in Disabled or Enabled>"}}
```

12. Network/Network Settings/Get Network Speed

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.Speed

Header: content-type application/json

Auth: Basic or X auth
```

13. Network/Network Settings/Set Network Speed

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"NIC.1.Speed": "<pass in 10, 100 or 1000>"}}
```

14. Network/Network Settings/Get Duplex Mode

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.Duplex

Header: content-type application/json

Auth: Basic or X auth
```

15. Network/Network Settings/Set Duplex Mode

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.Duplex": "<pass in Half or Full>"}}
```

16. Network/Network Settings/Get NIC MTU

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.MTU

Header: content-type application/json

Auth: Basic or X auth
```

17. Network/Network Settings/Set NIC MTU

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.MTU": "<pass in integer value from 576 to 1500>"}}
```

18. Network/iDRAC Auto Discovery/Get Auto Discovery Setting

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Autodiscovery.1.EnableIPChangeAnnounce

Header: content-type application/json

Auth: Basic or X auth
```

19. Network/iDRAC Auto Discovery/Set Auto Discovery Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Autodiscovery.1.EnableIPChangeAnnounce": "<pass in
Enabled or Disabled>"}}
```

20. Network/iDRAC Auto Discovery/Get Obtain Console Address Via DHCP

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Autodiscovery.1.EnableIPChangeAnnounceFromDHCP

Header: content-type application/json

Auth: Basic or X auth
```

21. Network/iDRAC Auto Discovery/Set Obtain Console Address Via DHCP

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Autodiscovery.1.EnableIPChangeAnnounceFromDHCP": "<pass
in Enabled or Disabled>"}}
```

22. Network/iDRAC Auto Discovery/Get Obtain Console Address Via Unicast DNS

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Autodiscovery.1.EnableIPChangeAnnounceFromUnicastDNS

Header: content-type application/json

Auth: Basic or X auth
```

23. Network/iDRAC Auto Discovery/Set Obtain Console Address Via Unicast DNS

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Autodiscovery.1.EnableIPChangeAnnounceFromUnicastDNS":
"<pass in Enabled or Disabled>"}}
```

24. Network/iDRAC Auto Discovery/Get Obtain Console Address Via mDNS

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Autodiscovery.1.EnableIPChangeAnnounceFrommDNS

Header: content-type application/json

Auth: Basic or X auth
```

25. Network/iDRAC Auto Discovery/Set Obtain Console Address Via mDNS

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Autodiscovery.1.EnableIPChangeAnnounceFrommDNS": "<pass
in Enabled or Disabled>"}}
```

26. Network/Common Settings/Get Register iDRAC on DNS

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.DNSRegister

Header: content-type application/json

Auth: Basic or X auth
```

27. Network/Common Settings/Set Register iDRAC on DNS

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.DNSRegister": "<pass in Enabled or Disabled>"}}
```

28. Network/Common Settings/Get DNS iDRAC Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.DNSRacName

Header: content-type application/json

Auth: Basic or X auth
```

29. Network/Common Settings/Set DNS iDRAC Name

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"NIC.1.DNSRacName": "<pass in string value>"}}
```

30. Network/Common Settings/Get Auto Config Domain Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.DNSDomainFromDHCP

Header: content-type application/json

Auth: Basic or X auth
```

31. Network/Common Settings/Set Auto Config Domain Name

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.DNSDomainFromDHCP": "<pass in Enabled or
Disabled>"}}
```

32. Network/Common Settings/Get Static Domain Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.DNSDomainName

Header: content-type application/json

Auth: Basic or X auth
```

33. Network/Common Settings/Set Static Domain Name

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.DNSDomainName": "<pass in string value>"}}
```

34. Network/Common Settings/Get DNS Register Interval

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.DNSRegisterInterval

Header: content-type application/json

Auth: Basic or X auth
```

35. Network/Common Settings/Set DNS Register Interval

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.DNSRegisterInterval": <pass in integer value from
60 to 7776000>}}
```

36. Network/Common Settings/Get Connection View

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SwitchConnectionView.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

37. Network/Common Settings/Set Connection View

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SwitchConnectionView.1.Enable": "<pass in Enabled or
Disabled>"}}
```

38. Network/Common Settings/Get Topology LLDP

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.TopologyLldp

Header: content-type application/json

Auth: Basic or X auth
```

39. Network/Common Settings/Set Topology LLDP

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1
```

```
Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.TopologyLldp": "<pass in Enabled or Disabled>"}}
```

40. Network/Common Settings/Get iDRAC Discovery LLDP

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.DiscoveryLLDP

Header: content-type application/json

Auth: Basic or X auth
```

41. Network/Common Settings/Set iDRAC Discovery LLDP

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.DiscoveryLLDP": "<pass in Enabled or Disabled>"}}
```

42. Network/Auto Config/Get DHCP Provisioning

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.AutoConfig

Header: content-type application/json

Auth: Basic or X auth
```

43. Network/Auto Config/Set DHCP Provisioning

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.AutoConfig": "<pass in Disabled, Enable Once, or
Enable Once After Reset>"}}
```

44. Network/IPv4 Settings/Get Enable IPv4

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

45. Network/IPv4 Settings/Set Enable IPv4

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4.1.Enable": "<pass in Disabled or Enabled>"}}
```

46. Network/IPv4 Settings/Get DHCP

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4.1.DHCPEnable

Header: content-type application/json

Auth: Basic or X auth
```

47. Network/IPv4 Settings/Set DHCP

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4.1.DHCPEnable": "<pass in Disabled or Enabled>"}}
```

48. Network/IPv4 Settings/Get Static IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4Static.1.Address

Header: content-type application/json

Auth: Basic or X auth
```

49. Network/IPv4 Settings/Set Static IP Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1
```

```
Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4Static.1.Address": "<pass in static IP address>"}}
```

50. Network/IPv4 Settings/Get Static Gateway

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4Static.1.Gateway

Header: content-type application/json

Auth: Basic or X auth
```

51. Network/IPv4 Settings/Set Static Gateway

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4Static.1.Gateway": "<pass in static gateway>"}}
```

52. Network/IPv4 Settings/Get Static Subnet Mask

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4Static.1.Netmask

Header: content-type application/json

Auth: Basic or X auth
```

53. Network/IPv4 Settings/Set Static Subnet Mask

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4Static.1.Netmask": "<pass in static netmask>"}}
```

54. Network/IPv4 Settings/Get Use DHCP to Obtain DNS Server Addresses

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4.1.DNSFromDHCP

Header: content-type application/json

Auth: Basic or X auth
```

55. Network/IPv4 Settings/Set Use DHCP to Obtain DNS Server Addresses

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4.1.DNSFromDHCP": "<pass in Enabled or Disabled>"}}
```

56. Network/IPv4 Settings/Get Static Preferred DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4Static.1.DNS1

Header: content-type application/json

Auth: Basic or X auth
```

57. Network/IPv4 Settings/Set Static Preferred DNS Server

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv4Static.1.DNS1": "<pass in static DNS 1 IP>"}}
```

58. Network/IPv4 Settings/Get Static Alternative DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv4Static.1.DNS2

Header: content-type application/json

Auth: Basic or X auth
```

59. Network/IPv4 Settings/Set Static Alternative DNS Server

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"IPv4Static.1.DNS2": "<pass in static DNS 2 IP>"}}
```

60. Network/IPv6 Settings/Get Enabled IPv6

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

61. Network/IPv6 Settings/Set Enabled IPv6

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6.1.Enable": "<pass in Enabled or Disabled>"}}
```

62. Network/IPv6 Settings/Get Address Generation Mode

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.AddressGenerationMode

Header: content-type application/json

Auth: Basic or X auth
```

63. Network/IPv6 Settings/Set Address Generation Mode

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6.1.AddressGenerationMode": "<pass in possible
value>"}}
```

64. Network/IPv6 Settings/Get Autoconfiguration

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.AutoConfig

Header: content-type application/json

Auth: Basic or X auth
```

65. Network/IPv6 Settings/Set Autoconfiguration

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6.1.AutoConfig": "<pass in possible value>"}}
```

66. Network/IPv6 Settings/Get Static IP Address 1

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6Static.1.Address1

Header: content-type application/json

Auth: Basic or X auth
```

67. Network/IPv6 Settings/Set Static IP Address 1

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6Static.1.Address1": "<pass in possible value>"}}
```

68. Network/IPv6 Settings/Get Static Prefix Length

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.PrefixLength

Header: content-type application/json

Auth: Basic or X auth
```

69. Network/IPv6 Settings/Set Static prefix Length

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"IPv6.1.PrefixLength": "<pass in possible value>"}}
```

70. Network/IPv6 Settings/Get Current IP Gateway

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6Static.1.Gateway

Header: content-type application/json

Auth: Basic or X auth
```

71. Network/IPv6 Settings/Set Current IP Gateway

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6Static.1.Gateway": "<pass in possible value>"}}
```

72. Network/IPv6 Settings/Get Link Local Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.LinkLocalAddress

Header: content-type application/json

Auth: Basic or X auth
```

73. Network/IPv6 Settings/Get Use DHCPv6 to obtain DNS Server Addresses

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.DNSFromDHCP6

Header: content-type application/json

Auth: Basic or X auth
```

74. Network/IPv6 Settings/Set Use DHCPv6 to obtain DNS Server Addresses

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6.1.DNSFromDHCP6": "<pass in possible value>"}}
```

75. Network/IPv6 Settings/Get Static Preferred DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.DNS1

Header: content-type application/json

Auth: Basic or X auth
```

76. Network/IPv6 Settings/Set Static Preferred DNS Server

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6.1.DNS1": "<pass in possible value>"}}
```

77. Network/IPv6 Settings/Get Static Alternate DNS Server

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPv6.1.DNS2

Header: content-type application/json

Auth: Basic or X auth
```

78. Network/IPv6 Settings/Set Static Alternate DNS Server

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPv6.1.DNS2": "<pass in possible value>"}}
```

79. Network/IPMI Settings/Get Enable IPMI Over LAN

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMILan.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

80. Network/IPMI Settings/Set Enable IPMI Over LAN

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMILan.1.Enable": "<pass in Enabled or Disabled>"}}
```

81. Network/IPMI Settings/Get Channel Privilege Level Limit

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMILan.1.PrivLimit

Header: content-type application/json

Auth: Basic or X auth
```

82. Network/IPMI Settings/Set Channel Privilege Level Limit

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1
```

```
Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMILan.1.PrivLimit": "<pass in User, Operator or
Administrator>"}}
```

83. Network/IPMI Settings/Get Encryption Key

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMILan.1.EncryptionKey

Header: content-type application/json

Auth: Basic or X auth
```

84. Network/IPMI Settings/Set Encryption Key

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMILan.1.EncryptionKey": "<pass in string value>"}}
```

85. Network/VLAN Settings/Get Enable VLAN ID

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.VLanEnable

Header: content-type application/json
```

```
Auth: Basic or X auth
```

86. Network/VLAN Settings/Set Enable VLAN ID

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.VLanEnable": "<pass in Enabled or Disabled>"}}
```

87. Network/VLAN Settings/Get VLAN ID

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.VLanID

Header: content-type application/json

Auth: Basic or X auth
```

88. Network/VLAN Settings/Set VLAN ID

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.VLanID": <pass in integer value from 1 to 4094>}}
```

89. Network/VLAN Settings/Get Priority

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NIC.1.VLanPriority

Header: content-type application/json

Auth: Basic or X auth
```

90. Network/VLAN Settings/Set Priority

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NIC.1.VLanPriority": <pass in integer from 0 to 7>}}
```

91. Network/Advanced Network Settings/IP Ranges/Get IP Range 1 Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.RangeEnable

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Same command can be used for IPBlocking.1.RangeEnable2, IPBlocking.1.RangeEnable3, IPBlocking.1.RangeEnable4 and IPBlocking.1.RangeEnable5.

92. Network/Advanced Network Settings/IP Ranges/Set IP Range 1 Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.RangeEnable": "<pass in possible value >"}}
```

NOTE: Same command can be used for IPBlocking.1.RangeEnable2, IPBlocking.1.RangeEnable3, IPBlocking.1.RangeEnable4 and IPBlocking.1.RangeEnable5.

93. Network/Advanced Network Settings/IP Ranges/Get IP Range 1 Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.RangeAddr

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Same command can be used for IPBlocking.1.RangeAddr2, IPBlocking.1.RangeAddr3, IPBlocking.1.RangeAddr4 and IPBlocking.1.RangeAddr5.

94. Network/Advanced Network Settings/IP Ranges/Set IP Range 1 Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.RangeAddr": "<pass in possible value >"}}
```

NOTE: Same command can be used for IPBlocking.1.RangeAddr2, IPBlocking.1.RangeAddr3, IPBlocking.1.RangeAddr4 and IPBlocking.1.RangeAddr5.

95. Network/Advanced Network Settings/IP Ranges/Get IP Range 1 Subnet Mask

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.RangeMask

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Same command can be used for IPBlocking.1.RangeMask2, IPBlocking.1.RangeMask3, IPBlocking.1.RangeMask4 and IPBlocking.1.RangeMask5.

96. Network/Advanced Network Settings/IP Ranges/Set IP Range 1 Subnet Mask

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.RangeMask": "<pass in possible value >"}}
```

NOTE: Same command can be used for IPBlocking.1.RangeMask2, IPBlocking.1.RangeMask3, IPBlocking.1.RangeMask4 and IPBlocking.1.RangeMask5.

97. Network/Advanced Network Settings/IP Blocking/Get IP Blocking Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.BlockEnable

Header: content-type application/json

Auth: Basic or X auth
```

98. Network/Advanced Network Settings/IP Blocking/Set IP Blocking Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.BlockEnable": "<pass in Enabled or
Disabled>"}}
```

99. Network/Advanced Network Settings/IP Blocking/Get IP Blocking Fail Count

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.FailCount

Header: content-type application/json

Auth: Basic or X auth
```

100. Network/Advanced Network Settings/IP Blocking/Set IP Blocking Fail Count

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.FailCount": <pass in integer value from 2 to
16>}}
```

101. Network/Advanced Network Settings/IP Blocking/Get IP Blocking Fail Window

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.FailWindow

Header: content-type application/json

Auth: Basic or X auth
```

102. Network/Advanced Network Settings/IP Blocking/Set IP Blocking Fail Window

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.FailWindow": <pass in integer value from 10
to 65535>}}
```

103. Network/Advanced Network Settings/IP Blocking/Get IP Blocking Penalty Time

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPBlocking.1.PenaltyTime

Header: content-type application/json

Auth: Basic or X auth
```

104. Network/Advanced Network Settings/Federal Information Processing Standards/Set IP Blocking Penalty Time

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPBlocking.1.PenaltyTime": <pass in integer value from 2
to 65535>}}
```

105. Network/Advanced Network Settings/Federal Information Processing Standards/Get FIPS Mode

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Security.1.FIPSMode

Header: content-type application/json

Auth: Basic or X auth
```

106. Network/Advanced Network Settings/Federal Information Processing Standards/Set FIPS Mode

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Security.1.FIPSMode": "<pass in Enabled or Disabled>"}}
```

107. Serial Over LAN/Get Enable Serial Over LAN

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMISOL.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

108. Serial Over LAN/Set Enable Serial Over LAN

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMISOL.1.Enable": "<pass in Enabled or Disabled>"}}
```

109. Serial Over LAN/Get Baud Rate

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMISOL.1.BaudRate

Header: content-type application/json

Auth: Basic or X auth
```

110. Serial Over LAN/Set Baud Rate

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMISOL.1.BaudRate": "<pass in 9600, 19200, 57600 or
115200 as a string value>"}}
```

111. Serial Over LAN/Get Channel Privilege Level Limit

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/IPMISOL.1.MinPrivilege

Header: content-type application/json

Auth: Basic or X auth
```

112. Serial Over LAN/Set Channel Privilege Level Limit

```
Command: PATCH
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"IPMISOL.1.MinPrivilege": "<pass in User, Operator or
Administrator>"}}
```

113. Serial Over LAN/Get Redirect Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SerialRedirection.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

114. Serial Over LAN/Set Redirect Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SerialRedirection.1.Enable": "<pass in Enabled or
Disabled>"}}
```

115. Serial Over LAN/Get Escape Key

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SerialRedirection.1.QuitKey
```

```
Header: content-type application/json

Auth: Basic or X auth
```

116. Serial Over LAN/Set Escape Key

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SerialRedirection.1.QuitKey": "<pass in string value>"}}
```

117. Operating system to iDRAC Pass-through/Pass-through Configuration/Get State

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OS-BMC.1.AdminState

Header: content-type application/json

Auth: Basic or X auth
```

118. Operating system to iDRAC Pass-through/Pass-through Configuration/Set State

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"OS-BMC.1.AdminState": "<pass in Enabled or Disabled>"}}
```

119. Operating system to iDRAC Pass-through/Pass-through Configuration/Get Pass-through Mode

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OS-BMC.1.PTMode

Header: content-type application/json

Auth: Basic or X auth
```

120. Operating system to iDRAC Pass-through/Pass-through Configuration/Set Pass-through Mode

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"OS-BMC.1.PTMode": "<pass in lom-p2p or usb-p2p>"}}
```

121. Operating system to iDRAC Pass-through/Network Settings/Get operating system IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OS-BMC.1.OsIpAddress

Header: content-type application/json

Auth: Basic or X auth
```

122. Operating system to iDRAC Pass-through/Network Settings/Set operating system IP Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"OS-BMC.1.OsIpAddress": "<pass in possible value >"}}
```

123. Operating system to iDRAC Pass-through/Network Settings/Get USB NIC IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OS-BMC.1.UsbNicIpAddress

Header: content-type application/json

Auth: Basic or X auth
```

124. Operating system to iDRAC Pass-through/Network Settings/Set USB NIC IP Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"OS-BMC.1.UsbNicIpAddress": "<pass in possible value >"}}
```

33 Section 33: iDRAC Settings/Services

1. Local Configuration/Get Disable iDRAC Local Configuration using Settings

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/LocalSecurity.1.PrebootConfig

Header: content-type application/json

Auth: Basic or X auth
```

2. Local Configuration/Set Disable iDRAC Local Configuration using Settings

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"LocalSecurity.1.PrebootConfig": "<pass in Enabled or
Disabled>"}}
```

3. Local Configuration/Get Disable iDRAC Local Configuration using RACADM

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/LocalSecurity.1.LocalConfig

Header: content-type application/json
```

```
Auth: Basic or X auth
```

4. Local Configuration/Set Disable iDRAC Local Configuration using RACADM

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"LocalSecurity.1.LocalConfig": "<pass in Enabled or
Disabled>"}}
```

5. Web Server/Settings/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

6. Web Server/Settings/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.Enable": "<pass in Enabled or Disabled>"}}
```

7. Web Server/Settings/Get HTTP/2 Protocol

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.Http2Enable

Header: content-type application/json

Auth: Basic or X auth
```

8. Web Server/Settings/Set HTTP/2 Protocol

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.Http2Enable": "<pass in Enabled or
Disabled>"}}
```

9. Web Server/Settings/Get Max Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.MaxNumberOfSessions

Header: content-type application/json

Auth: Basic or X auth
```

10. Web Server/Settings/Get Active Sessions

```
Command: GET
```

```
URI: redfish/v1/SessionService/Sessions?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Parse the JSON output and look for property SessionType with value of “WebUI”.

11. Web Server/Settings/Get Timeout

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.Timeout

Header: content-type application/json

Auth: Basic or X auth
```

12. Web Server/Settings/Set Timeout

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.Timeout": <pass in integer value from 60 to
10800>}}
```

13. Web Server/Settings/Get Block HTTP Port

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.BlockHTTPPort
```

```
Header: content-type application/json

Auth: Basic or X auth
```

14. Web Server/Settings/Set Block HTTP Port

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.BlockHTTPPort": "<pass in Enabled or
Disabled>"}}
```

15. Web Server/Settings/Get HTTPS Redirection

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.HttpsRedirection

Header: content-type application/json

Auth: Basic or X auth
```

16. Web Server/Settings/Set HTTPS Redirection

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.HttpsRedirection": "<pass in Enabled or
Disabled>"}}
```

17. Web Server/Settings/Get HTTP Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.HttpPort

Header: content-type application/json

Auth: Basic or X auth
```

18. Web Server/Settings/Set HTTP Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.HttpPort": <pass in integer value from 1 to
65535>}}
```

19. Web Server/Settings/Get HTTPS Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.HttpsPort

Header: content-type application/json

Auth: Basic or X auth
```

20. Web Server/Settings/Set HTTPS Port Number

```
Command: PATCH
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.HttpsPort": <pass in integer value from 1 to
65535>}}
```

21. Web Server/Settings/Get SSL Encryption

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.SSLEncryptionBitLength

Header: content-type application/json

Auth: Basic or X auth
```

22. Web Server/Settings/Set SSL Encryption

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.SSLEncryptionBitLength": "<pass in Auto-
Negotiate, 128-Bit or higher, 168-Bit or higher or 256-Bit or higher>"}}
```

23. Web Server/Settings/Get TLS Protocol

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.TLSProtocol
```

```
Header: content-type application/json

Auth: Basic or X auth
```

24. Web Server/Settings/Set TLS Protocol

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"WebServer.1.TLSProtocol": "<pass in TLS 1.1 and Higher,
TLS 1.2 Only, TLS 1.2 and Higher or TLS 1.3 Only>"}}
```

25. Web Server/Settings/Get Custom Cipher String

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/WebServer.1.CustomCipherString

Header: content-type application/json

Auth: Basic or X auth
```

26. Web Server/Settings/Set Custom Cipher String

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"WebServer.1.CustomCipherString": "<pass in possible
value>"}}
```

27. SSL/TLS Certificate Signing Request/Get Certs

```
Command: GET

URI: redfish/v1/CertificateService/CertificateLocations?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

28. SSL/TLS Certificate Signing Request/Generate CSR

```
Command: POST

URI: redfish/v1/CertificateService/Actions/CertificateService.GenerateCSR

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateCollection": {"@odata.id":
"/redfish/v1/Managers/iDRAC.Embedded.1/NetworkProtocol/HTTPS/Certificates"},
"City": "<your city>", "CommonName": "<your common name>", "Country": "<your
country>", "Organization": "<org name>", "OrganizationalUnit": "<org unit>",
"State": "<your state>", "email": "<your email>"}
```

29. SSL/TLS Certificate Signing Request/Download Cert

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRAC
CardService.ExportSSLCertificate

Header: content-type application/json

Auth: Basic or X token


Body: {"SSLCertType":"<cert type to download>"}

Body Example: {"SSLCertType":"Server"}
```

30. SSL/TLS Certificate Signing Request/Upload Cert

```
Command: POST

URI:
redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRAC
CardService.ImportSSLCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateType":"<cert type to upload>","SSLCertificateFile":"<pass in
cert as base64 string format>"}
```

31. SSL/TLS Certificate Signing Request/CA Certificate/Upload CA Certificate

```
Command: POST

URI:
/redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRA
CCardService.ImportCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateType":" SCEP_CA_CERT","CertificateFile"::"<cert in base64
string format>"}
```

32. SSH/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SSH.1.Enable

Header: content-type application/json
```

```
Auth: Basic or X auth
```

33. SSH/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SSH.1.Enable": "<pass in Enabled or Disabled>"}}
```

34. SSH/Get Max Sessions

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SSH.1.MaxSessions

Header: content-type application/json

Auth: Basic or X auth
```

35. SSH/Get Active Sessions

```
Command: GET

URI: redfish/v1/SessionService/Sessions?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Parse the JSON output and look for property SessionType with value of “ManagerConsole”.

36. SSH/Get Timeout

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SSH.1.Timeout

Header: content-type application/json

Auth: Basic or X auth
```

37. SSH/Set Timeout

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SSH.1.Timeout": <pass in integer value from 60 to
10800>}}
```

38. SSH/Get Port

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SSH.1.Port

Header: content-type application/json

Auth: Basic or X auth
```

39. SSH/Set Port

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SSH.1.Port": <pass in integer value from 1 to 65535>}}
```

40. Remote RACADM/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Racadm.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

41. Remote RACADM/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Racadm.1.Enable": "<pass in Enabled or Disabled>"}}
```

42. Remote RACADM/Get Active Sessions

```
Command: GET

URI: redfish/v1/SessionService/Sessions?$expand=*($levels=1)

Header: content-type application/json
```

```
Auth: Basic or X auth
```

NOTE: Parse the JSON output and look for the property SessionType with value of “OEM” and OemSessionType value of “racadm” to get all remote RACADM sessions.

43. SNMP Agent/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMP.1.AgentEnable

Header: content-type application/json

Auth: Basic or X auth
```

44. SNMP Agent/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMP.1.AgentEnable": "<pass in Enabled or Disabled>"}}
```

45. SNMP Agent/Get SNMP Community Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMP.1.AgentCommunity

Header: content-type application/json

Auth: Basic or X auth
```

46. SNMP Agent/Set SNMP Community Name

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMP.1.AgentCommunity": "<pass in possible string
value>"}}
```

47. SNMP Agent/Get SNMP Protocol

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMP.1.SNMPProtocol

Header: content-type application/json

Auth: Basic or X auth
```

48. SNMP Agent/Set SNMP Protocol

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMP.1.SNMPProtocol": "<pass in All or SNMPv3>"}}
```

49. SNMP Agent/Get SNMP Discovery Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SNMP.1.DiscoveryPort

Header: content-type application/json

Auth: Basic or X auth
```

50. SNMP Agent/Set SNMP Discovery Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SNMP.1.DiscoveryPort": <pass in integer value from 1 to
65535>}}
```

51. Automated System Recovery Agent/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ASRConfig.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

52. Automated System Recovery Agent/Set Enabled

```
Command: PATCH
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ASRConfig.1.Enable": "<pass in Enabled or Disabled>"}}
```

53. Redfish/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Redfish.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

54. Redfish/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Redfish.1.Enable": "<pass in Enabled or Disabled>"}}
```

34 Section 34: iDRAC Settings/Users

1. Local Users/Get iDRAC User Accounts

```
Command: GET

URI: redfish/v1/Managers/iDRAC.Embedded.1/Accounts?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

2. Local Users/Create New iDRAC User

```
Command: PATCH

URI: redfish/v1/Managers/iDRAC.Embedded.1/Accounts/{user account ID, value is 2
to 16}.

Header: content-type application/json

Auth: Basic or X auth

Body: {"UserName": "<new username string>", "Password": "<new user password>",
"RoleId": "<Supported values: Administrator, Operator, ReadOnly, or None>",
"Enabled": true}
```

3. Local Users/Change iDRAC User Account Password

```
Command: PATCH

URI: redfish/v1/Managers/iDRAC.Embedded.1/Accounts/{user account ID, value is 2
to 16}.

Header: content-type application/json

Auth: Basic or X auth

Body: {"Password": "<new user password>"}
```

4. Local Users/Change iDRAC Account User Privilege

```
Command: PATCH

URI: redfish/v1/Managers/iDRAC.Embedded.1/Accounts/{user account ID, value is 2
to 16}.

Header: content-type application/json

Auth: Basic or X auth

Body: {"RoleId": "<Supported values: Administrator, Operator, ReadOnly, or
None>"}
```

5. Local Users/Change iDRAC Account IPMI LAN Privilege

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.IpmiLanPrivilege": "<pass in User, Operator,
Administrator or No Access>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

6. Local Users/Change iDRAC Account IPMI Serial Port Privilege

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.IpmiSerialPrivilege": "<pass in User, Operator,
Administrator or No Access>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

7. Local Users/Change iDRAC Account IPMI Serial Over LAN Privilege

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.SolEnable": "<pass in Enabled or Disabled>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

8. Local Users/Change iDRAC Account SNMP v3 Enabled Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.ProtocolEnable": "<pass in Enabled or
Disabled>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

9. Local Users/Change iDRAC Account SNMP v3 Authentication Type

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"Users.2.AuthenticationProtocol": "<pass in possible
value>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

10. Local Users/Change iDRAC Account SNMP v3 Privacy Type

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.PrivacyProtocol": "<pass in None, DES or AES>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

11. Local Users/Change iDRAC Account SNMP v3 Enable Passphrase

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.EnableSNMPv3Passphrase": "<pass in possible
value>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

12. Local Users/Change iDRAC Account SNMP v3 Authentication Passphrase

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.SNMPv3AuthenticationPassphrase": "<pass in
possible value>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

13. Local Users/Change iDRAC Account SNMP v3 Privacy Passphrase

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.SNMPv3PrivacyPassphrase": "<pass in possible
value>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

14. Local Users/Change iDRAC Account Easy 2FA

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"Users.2.Simple2FA": "<pass in Enabled or Disabled>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

15. Local Users/Change iDRAC Account RSA Secure ID

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Users.2.RSASecurID2FA": "<pass in Enabled or Disabled>"}}
```

NOTE: Change “2” in the attribute name to the user ID index that you want to change.

16. Local Users/Get SSH Key

```
Command: GET

URI: redfish/v1/AccountService/Accounts/%s/Keys?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

17. Local Users/Upload SSH Key

```
Command: POST

URI: redfish/v1/AccountService/Accounts/{user account ID}/Keys

Header: content-type application/json

Auth: Basic or X token

Body: {"KeyType":"SSH", "KeyString":"(SSH key string)"}
```

18. Local Users/Delete SSH Key

Command: DELETE

URI: redfish/v1/AccountService/Accounts/{user account ID}/Keys/{key ID}

Header: content-type application/json

Auth: Basic or X token

19. Local Users/Get Smart Card User Certificate

Command: GET

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=SecurityCertificate.*

Header: content-type application/json

Auth: Basic or X auth

NOTE: Filter SecurityCertificate instance with "CertificateType" property as "SMARTCARD_USER".

20. Local Users/Upload Smart Card User Certificate

Command: POST

URI:
/redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRACCardService.ImportCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateType":"SMARTCARD_USER_CERT","Instance":<User Id index that you want to upload>,"CertificateFile"::"<cert in base64 string format>"}

21. Local Users/Get Smart Card Trusted CA Certificate

Command: GET

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$select=SecurityCertificate.*

Header: content-type application/json

Auth: Basic or X auth

NOTE: Filter SecurityCertificate instance with “CertificateType” property as “SMARTCARD_CA”.

22. Local Users/Upload Smart Card Trusted CA Certificate

Command: POST

URI:
/redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRACCardService.ImportCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateType":"SMARTCARD_CA_CERT",”Instance”:<User Id index that you want to upload>,"CertificateFile"::"<cert in base64 string format>"}

23. Local Users/Delete iDRAC User Account

Command: PATCH

URI: redfish/v1/Managers/iDRAC.Embedded.1/Accounts/{user account ID, value is 2 to 16}.

Header: content-type application/json

Auth: Basic or X auth

Body: {"Enabled":False,"RoleId":"None"}

Command: PATCH

URI: redfish/v1/Managers/iDRAC.Embedded.1/Accounts/{user account ID, value is 2 to 16}.

Header: content-type application/json

```
Auth: Basic or X auth

Body: {"UserName":""}
```

NOTE: Run (2) PATCH calls to completely remove this user ID entry from the iDRAC.

24. Directory Services/Microsoft Active Directory/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ActiveDirectory.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

25. Directory Services/Microsoft Active Directory/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ActiveDirectory.1.Enable": "<pass in Enabled or
Disabled>"}}
```

26. Directory Services/Microsoft Active Directory/Configure Active Directory

NOTE: GET on URI “redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1” and filter for attributes with keyword “ActiveDirectory” or “ADGroup”. These attributes are used to configure Active Directory. You can leverage PATCH command to configure them (same command as above for enabling Active Directory).

NOTE: If you are configuring multiple servers for AD, recommended to use the UI to configure one server. Next, perform “SCP clone export” for iDRAC attributes. Then, use this SCP file to push AD changes to multiple other servers.

27. Directory Services/Generic LDAP Directory Service/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/LDAP.1.Enable

Header: content-type application/json

Auth: Basic or X auth
```

28. Directory Services/Generic LDAP Directory Service/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"LDAP.1.Enable": "<pass in Enabled or Disabled>"}}
```

29. Directory Services/Generic LDAP Directory Service/Configure LDAP

NOTE: Run GET on URI “redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1”. Next, filter for attributes with keyword “LDAP”. These attributes are used to configure LDAP which you can leverage PATCH command to configure them (same command as above for enabling LDAP).

NOTE: When configuring multiple servers for LDAP, use the UI to configure one server. Then perform SCP clone export for iDRAC attributes, and use this SCP file to push LDAP changes to multiple other servers.

30. Smart Card/Get Configure Smart Card Logon

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SmartCard.1.SmartCardLogonEnable

Header: content-type application/json

Auth: Basic or X auth
```

31. Smart Card/Set Configure Smart Card Logon

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SmartCard.1.SmartCardLogonEnable": "<pass in Enabled or
Disabled>"}}
```

32. Smart Card/Get Enable CRL Check for Smart Card Logon

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/SmartCard.1.SmartCardCRLEnable

Header: content-type application/json

Auth: Basic or X auth
```

33. Smart Card/Set Enable CRL Check for Smart Card Logon

```
Command: PATCH
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"SmartCard.1.SmartCardCRLEnable": "<pass in Enabled or
Disabled>"}}
```

34. Sessions/Get Active Sessions

```
Command: GET

URI: redfish/v1/SessionService/Sessions?$expand=*($levels=1)

Header: content-type application/json

Auth: Basic or X auth
```

35. Sessions/Delete Session

```
Command: DELETE

URI: redfish/v1/Sessions/{session ID to delete}

Header: content-type application/json

Auth: Basic or X token
```

36. OpenID Connect Configured Systems/Get Enabled

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OpenIDConnectServer.9.Enabled

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Replace “9” with your OpenID number.

37. OpenID Connect Configured Systems/Set Enabled

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"OpenIDConnectServer.9.Enabled": "<pass in Enabled or
Disabled>"}}
```

NOTE: Replace “9” with your OpenID number.

38. OpenID Connect Configured Systems/Get System Name

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OpenIDConnectServer.9.Name

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Replace “9” with your OpenID number.

39. OpenID Connect Configured Systems/Set System Name

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"OpenIDConnectServer.9.Name": "<pass in possible value>"}}
```

NOTE: Replace “9” with your OpenID number.

40. OpenID Connect Configured Systems/Get Discovery URL

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OpenIDConnectServer.9.DiscoveryURL

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Replace “9” with your OpenID number.

41. OpenID Connect Configured Systems/Set Discovery URL

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"OpenIDConnectServer.9.DiscoveryURL": "<pass in possible
value>"}}
```

NOTE: Replace “9” with your OpenID number.

42. OpenID Connect Configured Systems/Get Registration Status

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/OpenIDConnectServer.9.RegistrationStatus

Header: content-type application/json
```

```
Auth: Basic or X auth
```

NOTE: Replace “9” with your OpenID number.

43. Global User Settings/Password Settings/Get Default Password Warning

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/DefaultCredentialMitigationConfigGroup.1.DefaultCredentialMitig
ation

Header: content-type application/json

Auth: Basic or X auth
```

44. Global User Settings/Password Settings/Set Default Password Warning

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes":
{"DefaultCredentialMitigationConfigGroup.1.DefaultCredentialMitigation": "<pass
in possible value>"}}
```

45. Global User Settings/Password Settings/Policy Settings/Get Minimum Score

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Security.1.MinimumPasswordScore

Header: content-type application/json

Auth: Basic or X auth
```

46. Global User Settings/Password Settings/Policy Settings/Get Minimum Score

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Security.1.MinimumPasswordScore": "<pass in No
Protection, Weak Protection, Moderate Protection or Strong Protection>"}}
```

47. Global User Settings/Password Settings/Policy Settings/Get Simple Policy Upper Case Letters

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Security.1.PasswordRequireUpperCase

Header: content-type application/json

Auth: Basic or X auth
```

48. Global User Settings/Password Settings/Policy Settings/Set Simple Policy Upper Case Letters

```
Command: PATCH
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Security.1.PasswordRequireUpperCase": "<pass in Enabled
or Disabled>"}}
```

49. Global User Settings/Password Settings/Policy Settings/Get Simple Policy Numbers

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Security.1.PasswordRequireNumbers

Header: content-type application/json

Auth: Basic or X auth
```

50. Global User Settings/Password Settings/Policy Settings/Set Simple Policy Numbers

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Security.1.PasswordRequireNumbers": "<pass in Enabled or
Disabled>"}}
```

51. Global User Settings/Password Settings/Policy Settings/Get Simple Policy Symbols

```
Command: GET
```

```
URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Security.1.PasswordRequireSymbols

Header: content-type application/json

Auth: Basic or X auth
```

52. Global User Settings/Password Settings/Policy Settings/Set Simple Policy Symbols

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Security.1.PasswordRequireSymbols": "<pass in Enabled or
Disabled>"}}
```

53. Global User Settings/Password Settings/Policy Settings/Get Regular Expression

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Security.1.PasswordRequireRegex

Header: content-type application/json

Auth: Basic or X auth
```

54. Global User Settings/Password Settings/Policy Settings/Set Regular Expression

```
Command: PATCH
```

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Security.1.PasswordRequireRegex": "<pass in string value>"}}

35 Section 35: iDRAC Settings/Settings

1. Time Zone and NTP Settings/Time Zone Settings/Get Time Zone

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/Time.1.Timezone

Header: content-type application/json

Auth: Basic or X auth
```

2. Time Zone and NTP Settings/Time Zone Settings/Set Time Zone

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"Time.1.Timezone": "<pass in possible value>"}}
```

NOTE: String possible values will not be available in the attribute registry. Use iDRAC GUI page drop down menu for this attribute to get possible string values.

3. Time Zone and NTP Settings/NTP Server Settings/Get Enable Network Time Protocol (NTP)

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NTPConfigGroup.1.NTPEnable

Header: content-type application/json

Auth: Basic or X auth
```

4. Time Zone and NTP Settings/NTP Server Settings/Set Enable Network Time Protocol (NTP)

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NTPConfigGroup.1.NTPEnable": "<pass in Enabled or
Disabled>"}}
```

5. Time Zone and NTP Settings/NTP Server Settings/Get NTP Server 1

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/NTPConfigGroup.1.NTP1

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: Replace “1” with 2 or 3 to get NTP server details.

6. Time Zone and NTP Settings/NTP Server Settings/Set NTP Server 1

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"NTPConfigGroup.1.NTP1": "<pass in possible value>"}}
```

NOTE: Replace “1” with 2 or 3 to set that NTP server.

7. iDRAC Service Module Setup/Service Module Installation/Get Installation Status

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory/Installed-104684-
4.0.1__ServiceModule.Embedded.1?$select=Oem/Dell/DellSoftwareInventory/Status

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: If your embedded version is different, run GET on URI “redfish/v1/UpdateService/FirmwareInventory” first to get the version and replace “4.0.1” with your version in the URI.

8. iDRAC Service Module Setup/Service Module Installation/Get Version

```
Command: GET

URI: redfish/v1/UpdateService/FirmwareInventory/Installed-104684-
4.0.1__ServiceModule.Embedded.1?$select=Version

Header: content-type application/json

Auth: Basic or X auth
```

NOTE: If your embedded version is different, run GET on URI “redfish/v1/UpdateService/FirmwareInventory” first to get the version and replace “4.0.1” with your version in the URI.

9. iDRAC Service Module Setup/Version/Get Installed Version on Host operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.ServiceModuleVersion

Header: content-type application/json
```

```
Auth: Basic or X auth
```

10. iDRAC Service Module Setup/Service Module Status/Get Connection Status on Host operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.ServiceModuleState

Header: content-type application/json

Auth: Basic or X auth
```

11. iDRAC Service Module Setup/Service Module Status/Get Service on Host operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.ServiceModuleEnable

Header: content-type application/json

Auth: Basic or X auth
```

12. iDRAC Service Module Setup/Service Module Status/Set Service on Host operating system

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"ServiceModule.1.ServiceModuleEnable": "<pass in Enabled
or Disabled>"}}
```

13. iDRAC Service Module Setup/Monitoring/Get operating system Information

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.OSInfo

Header: content-type application/json

Auth: Basic or X auth
```

14. iDRAC Service Module Setup/Monitoring/Set operating system Information

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.OSInfo": "<pass in Enabled or
Disabled>"}}
```

15. iDRAC Service Module Setup/Monitoring/Get Replicate Lifecycle Login operating system Log

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.LCLReplication

Header: content-type application/json
```

```
Auth: Basic or X auth
```

16. iDRAC Service Module Setup/Monitoring/Set Replicate Lifecycle Login operating system Log

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.LCLReplication": "<pass in Enabled or
Disabled>"}}
```

17. iDRAC Service Module Setup/Monitoring/Get WMI Information

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.WMIInfo

Header: content-type application/json

Auth: Basic or X auth
```

18. iDRAC Service Module Setup/Monitoring/Set WMI Information

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.WMIInfo": "<pass in Enabled or
Disabled>"}}
```

19. iDRAC Service Module Setup/Monitoring/Get Auto System Recovery

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.WatchdogState

Header: content-type application/json

Auth: Basic or X auth
```

20. iDRAC Service Module Setup/Monitoring/Set Auto System Recovery

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.WatchdogState": "<pass in Enabled or
Disabled>"}}
```

21. iDRAC Service Module Setup/Monitoring/Get Auto System Recovery Action

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.WatchdogRecoveryAction

Header: content-type application/json

Auth: Basic or X auth
```

22. iDRAC Service Module Setup/Monitoring/Set Auto System Recovery Action

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.WatchdogRecoveryAction": "<pass None,
Reboot, Poweroff or Powercycle>"}}
```

23. iDRAC Service Module Setup/Monitoring/Get Allow Service Module to perform iDRAC Hard Reset

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.iDRACHardReset

Header: content-type application/json

Auth: Basic or X auth
```

24. iDRAC Service Module Setup/Monitoring/Set Allow Service Module to perform iDRAC Hard Reset

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.iDRACHardReset": "<pass in Enabled or
Disabled>"}}
```

25. iDRAC Service Module Setup/Monitoring/Get Enable SNMP Alerts using Host operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.HostSNMPAlert

Header: content-type application/json

Auth: Basic or X auth
```

26. iDRAC Service Module Setup/Monitoring/Set Enable SNMP Alerts using Host operating system

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.HostSNMPAlert": "<pass in Enabled or
Disabled>"}}
```

27. iDRAC Service Module Setup/Monitoring/Get Enable SNMP OMSA Alerts using Host operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.HostSNMPOMSAAlert

Header: content-type application/json

Auth: Basic or X auth
```

28. iDRAC Service Module Setup/Monitoring/Set Enable SNMP OMSA Alerts using Host operating system

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.HostSNMPOMSAAlert": "<pass in Enabled or
Disabled>"}}
```

29. iDRAC Service Module Setup/Monitoring/Get Enable SNMP Get using Host operating system

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.HostSNMPGet

Header: content-type application/json

Auth: Basic or X auth
```

30. iDRAC Service Module Setup/Monitoring/Set Enable SNMP Get using Host operating system

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"ServiceModule.1.HostSNMPGet": "<pass in Enabled or
Disabled>"}}
```

31. iDRAC Service Module Setup/Monitoring/Get iDRAC SSO Launcher

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.iDRACSSOLauncher

Header: content-type application/json

Auth: Basic or X auth
```

32. iDRAC Service Module Setup/Monitoring/Set iDRAC SSO Launcher

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.iDRACSSOLauncher": "<pass in Enabled or
Disabled>"}}
```

33. iDRAC Service Module Setup/Monitoring/Get SDS Event Correlation

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.SSEventCorrelation

Header: content-type application/json
```

```
Auth: Basic or X auth
```

34. iDRAC Service Module Setup/Monitoring/Set SDS Event Correlation

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"ServiceModule.1.SSEventCorrelation": "<pass in NA,
Enabled or Disabled>"}}
```

35. iDRAC Service Module Setup/Monitoring/Get SATA Supported Chipset

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/ServiceModule.1.ChipsetSATASupported

Header: content-type application/json

Auth: Basic or X auth
```

36. iDRAC Service Module Setup/Monitoring/Set SATA Supported Chipset

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token
```

```
Body: {"Attributes": {"ServiceModule.1.ChipsetSATASupported": "<pass in NA,
Enabled or Disabled>"}}
```

37. Management USB Settings/Get USB Management Port Setting

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/USB.1.PortStatus

Header: content-type application/json

Auth: Basic or X auth
```

38. Management USB Settings/Set USB Management Port Setting

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"USB.1.PortStatus": "<pass in Enabled or Disabled>"}}
```

39. Management USB Settings/Get iDRAC Managed USB SCP

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/USB.1.ConfigurationXML

Header: content-type application/json

Auth: Basic or X auth
```

40. Management USB Settings/Set iDRAC Managed USB SCP

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"USB.1.ConfigurationXML": "<pass in Disabled, Enabled
while server has default credential settings only, Enabled or Enabled only for
compressed configuration files>"}}
```

41. Management USB Settings/Set Password for Zip file

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"USB.1.ZipPassword": "<pass in possible value>"}}
```

42. Management USB Settings/Get Device Present

```
Command: GET

URI:
/redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellUSBDevices?$select=Members@od
ata.count

Header: content-type application/json

Auth: Basic or X token
```

NOTE: Members@odata.count property value is 0 when USB device is absent otherwise it is present.

43. Alert Configuration/SMTP(email) Server Settings/Get SMTP Server IP Address

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPServerIPAddress

Header: content-type application/json

Auth: Basic or X token
```

44. Alert Configuration/SMTP(email) Server Settings/Set SMTP Server IP Address

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPServerIPAddress": "<new value you want
to set>"}}
```

45. Alert Configuration/SMTP(email) Server Settings/Get SMTP Port Number

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPPort

Header: content-type application/json

Auth: Basic or X token
```

46. Alert Configuration/SMTP(email) Server Settings/Set SMTP Port Number

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPPort": <pass in integer value from 1 to
65535>}}
```

47. Alert Configuration/SMTP(email) Server Settings/Get SMTP Authentication

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPAuthentication

Header: content-type application/json

Auth: Basic or X token
```

48. Alert Configuration/SMTP(email) Server Settings/Set SMTP Authentication

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPAuthentication": "<pass in Enabled or
Disabled>"}}
```

49. Alert Configuration/SMTP(email) Server Settings/Get SMTP Username

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPUserName

Header: content-type application/json

Auth: Basic or X token
```

50. Alert Configuration/SMTP(email) Server Settings/Set SMTP Username

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPUserName": "<new value you want to
set>"}}
```

51. Alert Configuration/SMTP(email) Server Settings/Get SMTP Username Password

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.SMTPPassword

Header: content-type application/json

Auth: Basic or X token
```

52. Alert Configuration/SMTP(email) Server Settings/Set SMTP Username Password

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.SMTPPassword": "<new value you want to
set>"}}
```

53. Alert Configuration/SMTP(email) Server Settings/Get SMTP Connection Encryption

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RemoteHosts.1.ConnectionEncryption

Header: content-type application/json

Auth: Basic or X token
```

54. Alert Configuration/SMTP(email) Server Settings/Set SMTP Connection Encryption

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RemoteHosts.1.ConnectionEncryption": "<pass in None,
SSL/TLS or STARTTLS>"}}
```

55. RSA SecurID Configuration/RSA Server Certificate/Upload RSA Server Certificate

```
Command: POST

URI:
/redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRA
CCardService.ImportCertificate

Header: content-type application/json

Auth: Basic or X token

Body: {"CertificateType":"RSA_CA_CERT","CertificateFile"::"<cert in base64
string format>"}
```

56. RSA SecurID Configuration/RSA Server Certificate/Test Network Connection

```
Command: POST

URI:
/redfish/v1/Dell/Managers/iDRAC.Embedded.1/DelliDRACCardService/Actions/DelliDRA
CCardService.TestConnection

Header: content-type application/json

Auth: Basic or X token

Body: {"ConnectionType":"RSA_SERVER"}
```

57. RSA SecurID Configuration/RSA SecurID Server Settings/Get RSA SecurID Authentication Server URL

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RSASecurID2FA.1.RSASecurIDAuthenticationServer

Header: content-type application/json
```

```
Auth: Basic or X token
```

58. Alert Configuration/SMTP(email) Server Settings/Set SMTP Connection Encryption

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RSASecurID2FA.1.RSASecurIDAuthenticationServer": "<new
value you want to set>"}}
```

59. RSA SecurID Configuration/RSA SecurID Server Settings/Get RSA SecurID Client ID

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RSASecurID2FA.1.RSASecurIDClientID

Header: content-type application/json

Auth: Basic or X token
```

60. Alert Configuration/SMTP(email) Server Settings/Set RSA SecurID Client ID

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json
```

```
Auth: Basic or X token

Body: {"Attributes": {"RSASecurID2FA.1.RSASecurIDClientID": "<new value you want
to set>"}}
```

61. RSA SecurID Configuration/RSA SecurID Server Settings/Get RSA SecurID Access Key

```
Command: GET

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1?$s
elect=Attributes/RSASecurID2FA.1.RSASecurIDAccessKey

Header: content-type application/json

Auth: Basic or X token
```

62. Alert Configuration/SMTP(email) Server Settings/Set RSA SecurID Access Key

```
Command: PATCH

URI:
redfish/v1/Managers/iDRAC.Embedded.1/Oem/Dell/DellAttributes/iDRAC.Embedded.1

Header: content-type application/json

Auth: Basic or X token

Body: {"Attributes": {"RSASecurID2FA.1.RSASecurIDAccessKey": "<new value you
want to set>"}}
```

36 Section 36: Attribute Registry

NOTE: Running GET or PATCH operations for iDRAC settings(attributes), recommended checking the attribute registry first for supported values, read-only, read writable or any dependencies. All registries are posted under URI “redfish/v1/Registries”.

1. Get iDRAC Attribute Registry

```
Command: GET

URI: redfish/v1/Registries/ManagerAttributeRegistry

Header: content-type application/json

Auth: Basic or X auth
```

2. Get BIOS Attribute Registry

```
Command: GET

URI: redfish/v1/Systems/System.Embedded.1/Bios/BiosRegistry

Header: content-type application/json

Auth: Basic or X auth
```

3. Get Network Attribute Registry

```
Command: GET

URI: /redfish/v1/Registries/NetworkAttributesRegistry_{network ID}

URI Example: /redfish/v1/Registries/NetworkAttributesRegistry_NIC.Embedded.1-1-1

Header: content-type application/json

Auth: Basic or X auth
```

37 Section 37: Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.