Public



AMD Manageability API Usage Guide

(Version 9.0)

Issue Date: October 2026

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS" AND AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS PUBLICATION AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT DESCRIPTIONS AT ANY TIME WITHOUT NOTICE.

The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication.

EXCEPT AS SET FORTH IN AMD'S STANDARD TERMS AND CONDITIONS OF SALE, AMD ASSUMES NO LIABILITY WHATSOEVER, AND DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT.

AMD's products are not designated, intended, authorized or warranted for use as components in systems intended for surgical implant in the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's products could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Trademarks

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Microsoft and Windows are registered trademarks of Microsoft Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

Copyright © 2026 Advanced Micro Devices, Inc. All rights reserved.

Table of Contents

Abbreviations ..... a
1 Introduction 1
1.1 Download 1
1.2 Installation 1
1.3 Configuration 1
1.4 Usage 2
1.4.1 Using Curl with a JSON file 2
1.4.2 Using Curl with a pipe 2
1.4.3 Using Postman tool 3
2 Sample classes for Input, Output & Error 5
2.1 Input class to generate JSON input 5
2.2 Output class used for different commands 6
2.3 Error class 6
3 Check Connection and Version 7
3.1 Check DASH System Connection 7
3.2 Check AMA Version 7
4 Discover 8
4.1 Discover Host 8
4.2 Discover Host with mutual cert 8
4.3 Discover IP 9
4.4 Discover Host Array 10
4.5 Discover IP Range 11
4.6 Discover Info 12
5 DASH Profiles 14
5.1 Enumerate Profile 14
5.2 Profile Show 14
5.3 Supported profiles 15
5.4 Enumerate All 16
6 ACMS 17
6.1 ACMS Show 17
7 Alert Destination 18
7.1 Alert Destination Show 18

7.2 Alert Destination Delete 18
8 Asset 19
8.1 Asset Show 19
9 Battery 20
9.1 Battery Show 20
9.2 Battery Enable 21
9.3 Battery Disable 21
9.4 Battery Test 21
9.5 Battery Recharge 21
10 BIOS 23
10.1 BIOS Show 23
10.2 BIOS Set Attribute 25
11 Boot Config 26
11.1 Boot Config Show 26
11.2 Boot Config Change Boot Order 26
12 Computer System 28
12.1 Computer System Show 28
12.2 Enumerate Asset 29
12.3 Enumerate Boot Config 30
12.4 Enumerate Fan 31
12.5 Enumerate Memory 33
12.6 Enumerate Power Supply 33
12.7 Enumerate Processor 34
12.8 Enumerate Sensor 35
12.9 Enumerate Software 38
12.10 Enumerate User 40
12.11 Power Status 41
12.12 Power Available States 41
12.13 Power Available States 2 41
12.14 Power Supported States 43
12.15 Power Supported States 2 43
12.16 Power State Change 45
12.17 Power State String 46

12.18 Power State Description 48
12.19 Asset 49
12.20 Boot Config 50
12.21 Fan 51
12.22 Memory 52
12.23 Power Supply 53
12.24 Processors 54
12.25 Sensor 55
12.26 Software 56
12.27 User Management 57
13 DHCP Client 59
13.1 DHCP Client Show 59
14 Disk Information 60
14.1 Disk Information Show 60
Disk Information wipedisk Actions 60
15 DNS Client 62
15.1 DNS Client Show 62
16 Ethernet Port 63
16.1 Ethernet Port Show 63
17 Fan 64
17.1 Fan Show 64
17.2 Enumerate Asset 64
17.3 Asset 67
18 Filter Collection 68
18.1 Filter Collection Show 68
18.2 Filter Collection Show All Filters 68
18.3 Filter Collection Delete 68
19 Indication 70
19.1 Indication Subscribe 70
20 Indication Filter 71
20.1 Indication Filter Show 71
20.2 Indication Filter Delete 71
21 Indication Subscription 72

21.1 Indication Subscription Show 72
22 Indicator LED 73
22.1 Indicator LED Show 73
23 IP Configuration 75
23.1 IP Configuration Show 75
24 IP Interface 77
24.1 IP Interface Show 77
25 KVM Redirection 78
25.1 KVM Redirection Show 78
25.2 KVM Redirection Actions 78
26 Log Entry 80
26.1 Log Entry Show 80
27 Media Redirection 81
27.1 Media Redirection Show 81
28 Memory 83
28.1 Memory Show 83
28.2 Enumerate Asset 83
28.3 Asset Show 85
29 Network Port 87
29.1 Network Port Show 87
30 Opaque Management Data 88
30.1 Opaque Management Data Show 88
31 Operating System 89
31.1 Operating System Show 89
32 PCI Device 90
32.1 PCI Device Show 90
33 Physical Computer System View 92
33.1 Physical Computer System View Show 92
34 Platform Watchdog 95
34.1 Platform Watchdog Show 95
35 Power Scheduler 97
35.1 Power Scheduler Show 97
35.2 Power Scheduler Add 97

35.3 Power Scheduler Clear 97
Power Scheduler Update 98
Power Scheduler Delete 98
36 Power Supply 99
36.1 Power Supply Show 99
36.2 Enumerate Asset 99
36.3 Asset 102
37 Processor 104
37.1 Processor Show 104
37.2 Enumerate Asset 105
37.3 Enumerate Fan 108
37.4 Enumerate Memory 109
37.5 Enumerate Sensor 111
37.6 Asset Show 112
37.7 Fan 113
37.8 Memory 113
37.9 Sensor Show 114
38 Record Log 116
38.1 Record Log Show 116
39 Registered Profile 117
39.1 Registered Profile Show 117
40 Roles 118
40.1 Role Show 118
41 Sensor 119
41.1 Sensor Show 119
41.2 Sensor Reading 120
41.3 Sensor Set Threshold 120
42 Service Processor 121
42.1 Service Processor Show 121
43 Software 122
43.1 Software Show 122
43.2 Software Install 122
44 Software Update 123

44.1 Software Update Start 123
45 SSH 124
45.1 SSH Show 124
46 Telnet 126
46.1 Telnet Show 126
47 Text Redirection 128
47.1 Text Redirection Show 128
47.2 Text Redirection Actions 128
48 USB Redirection 130
48.1 USB Redirection Show 130
48.2 USB Redirection Actions 130
49 User 133
49.1 User Show 133
49.2 User Add 133
49.3 User Enable 133
49.4 User Disable 134
49.5 User Assign Roles 134
49.6 User Remove Roles 134
49.7 User Delete 134
50 Wi-Fi Port 136
50.1 Wi-Fi-Port Show 136
51 Parse DASH Alert XML 138
51.1 Parse Alert 138
Appendix 140
A. HTTPS Sample commands 140
B. Logging 141
C. Error 142
D. Troubleshooting 149
D.1. Connection Issues 149
D.2. TLS / Certificate Issues 149
D.3. Cloud Relay (ACMS) Issues 150
D.4. Proxy Issues 150
D.5. Command and Argument Errors 151

D.6. Profile-Specific Issues 151
D.7. General Troubleshooting Tips 151
A. Support 153
Discussion forum link 153
DASH support email 153

Public

Abbreviations

Term	Definition
ACMS	AMD Cloud Manageability Service
AGESA	AMD Generic Encapsulated Software Architecture Interface Specification
AIM-T	AMD Integrated Management Technology
AMA	AMD Manageability API
AMC	AMD Management Console
AMD	Advanced Micro Devices
AMPS	AMD Management Plugin for SCCM
API	Application program interface
ASP	AMD Secure Processor
BDS	Boot Device Selection
BIOS	Basic Input/Output System
CLI	Command Line Interface
CRB	Customer Reference Board
CTR	Console Text Redirection
DASH	Desktop and Mobile Architecture for System Hardware
DMTF	Distributed Management Task Force
DXE	Driver Execution Environment
EC	Embedded Controller (BIOS)
FW	Firmware
GFX	Graphics
IT	Information Technology
KB	Keyboard
KVM	Keyboard Video and Mouse
LoM	LAN on Motherboard/Mainboard
MC	Management Controller
MCTP	Management Component Transfer Protocol
NIC	Network Interface Controller (On board Ethernet Controller)
OEM	Original Equipment Manufacturer
OS	Operating System (like Microsoft Windows, Linux)
PC	Personal Computer
PLDM	Platform Level Data Model
PXE	Preboot Execution Environment
RFB	Remote Frame Buffer
RGB	Red green blue
ROM	Read Only Memory
S0 - S5	ACPI Sleep States
S0ix	Standby power states
SBIOS	System BIOS
SDK	Software Development Kit
SMBIOS	System Management BIOS
SMBus	System Management Bus
SoC	System on Chip
SPI	Serial Peripheral Interface
SSH	Secure Shell
TCR	Text Console Redirection

TLS	Transport Layer Security
UEFI	Unified Extensible Firmware Interface
UI	User Interface
URI	Uniform Resource Identifier
USB	Universal Serial Bus
USB-R	USB Redirection
VNC	Virtual Network Computing
Wi-Fi	Wireless Fidelity
WLAN	Wireless LAN
WPA	Wi-Fi Protected Access
WS-MAN	Web Service Management

1 Introduction

AMD Manageability API (AMA) is a client manageability service from AMD to help IT administrators utilize DASH manageability by plugging in DASH capabilities to their management consoles. AMA takes JSON input and gives JSON output via REST APIs as depicted in Figure 1.

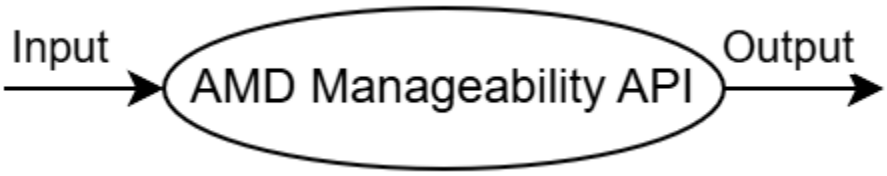


Figure 1: Depiction of JSON IO in AMA

AMA handles communication with the DASH system and responds back to the user JSON request with JSON response by abstracting the complex stack involved in the process.

1.1 Download

AMA can be downloaded from: www.amd.com/DASH

1.2 Installation

The download site links to an installer that can be downloaded by end users.
The default port is 8080, it can be changed during installation. If users want to change the port after installation, they can follow the steps from Configuration step in section 1.3.

1.3 Configuration

After installation the service starts and listens to the default chosen port. However, this port can be changed by the users to custom port by modifying the settings file which is placed by default at the location: ***C:\Program Files\AMD\Manageability API\config\settings.json***

Field	**Subfield**	**Default Value**	**Comments**
address		0.0.0.0	To use on localhost: 127.0.0.1 To listen on all interfaces: 0.0.0.0 To listen to specific IP, change to the IP
Port		8080	Port used by AMD Manageability API to listen to incoming requests
max_connections		512	Number of parallel incoming API requests
pem_file			Certificate section includes cert and private_key.
	cert	certs\\cert.pem	cert field should have relative path to the certificate that can be used for HTTPS communication.
	private_key	certs\\cert.pem	private_key should have the certificate key used for HTTPS communication.
ssh_proxy			SSH Proxy

	keyfile	““	Key to the SSH Proxy

Authentication and Security Guidance: AMA supports digest authentication via username/password in the JSON payload. For production use, enable HTTPS by configuring the certificate fields. Avoid hard-coding credentials in scripts; use environment variables or secure vaults. Mutual TLS is supported for all DASH operations.

1.4 Usage

This section explains how to interact with the AMD Manageability API (AMA) via REST API calls. AMA accepts JSON-formatted requests via HTTPS POST and returns structured JSON responses, enabling seamless integration with management consoles, automation scripts, and custom applications.

Protocol	HTTPS
Method	POST
URL	https://<AMA_HOST>:<AMA_PORT>/
Examples	https://127.0.0.1:8080/ https://10.252.220.35:8080/ https://DASH-USER:8080/

Default URI: https://127.0.0.1:8080

Input JSON	`{"h":"dash-system","Commands":["discover"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/DiscoveryCollection", "@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection", "Members" : [{ "@odata.id" : "/DASH/v1/DiscoveryCollection/Discover", "@odata.type" : "#Discover.v1_0_0.Discover", "DashPort" : 623, "Discovered" : true, "HostName" : "dash-system", "Name" : "Discover" }], "Name" : "Discovery Collection" }`

The corresponding Curl commands are explained in section 1.4.1 and 1.4.2. And the corresponding Postman command is explained in section 1.4.3.

1.4.1 Using Curl with a JSON file

>> curl -k -d @payload.json https://127.0.0.1:8080

Where the command expects payload.json to be present in the same path:

payload.json	`{ "h": "dash-system", "Commands": ["discover"] }`

Throughout the rest of the document, users can copy the Input JSON into the payload.json file to see the Output JSON on the screen by running the Curl command above.

1.4.2 Using Curl with a pipe

>> echo {“h”:”dash-system”, “u”:”admin”, “P”:”adminpass”,”C”:1,”Commands”:[“discover”]} | curl -k -d @- https://127.0.0.1:8080

1.4.2.1 Payload

In this command the payload for discovery includes username (u), password (P) and an option to ignore certificate verification (C).

Payload	<pre>{
	  "h": "dash-system",
	  "u": "admin",
	  "P": "adminpass",
	  "C": 1,
	  "Commands": ["discover"]
	}</pre>

1.4.2.2 echo Command

- Generates the JSON payload inline.
- Avoids the need for a separate file.

1.4.2.3 Pipe (|)

- Passes the output of echo directly as input to curl.

1.4.2.4 curl Options

- -k: Allows insecure HTTPS connections (skips certificate validation).
- -d: Data to be sent in the HTTP request body.
- @-: Reads the request body from standard input (stdin), which comes from the echo command.
    - @ symbol means: read data from a source.
    - - (dash) means: read from standard input (stdin).
- URL (https://127.0.0.1:8080): Target REST endpoint where the DASH service is listening.

1.4.3 Using Postman tool

For more advanced users familiar with API testing, the Postman application can be given the inputs in the POST query with raw JSON input to get the output JSON in the Output prompt as shown in the Figure 2.

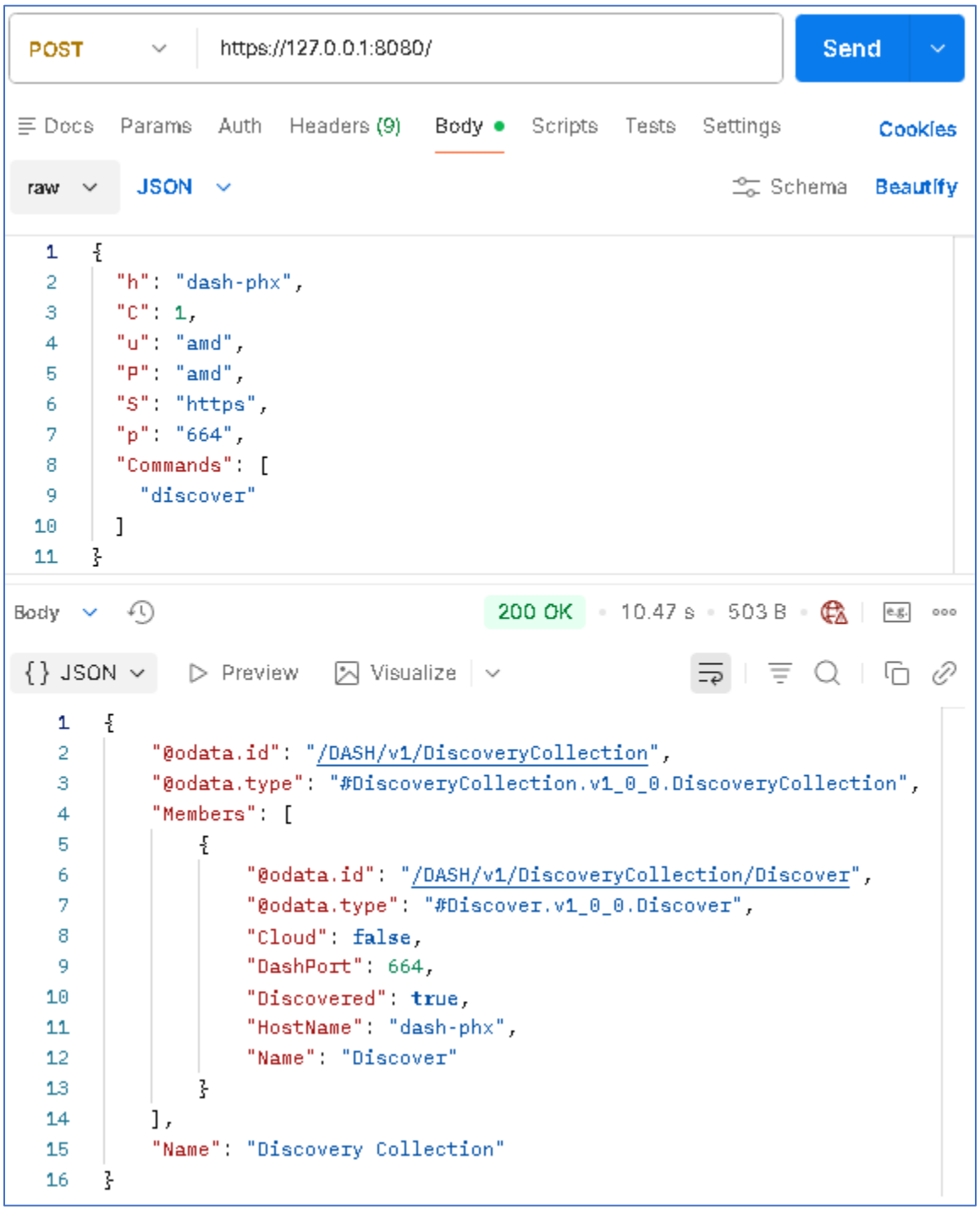


Figure 2: Depiction of AMA Discovery query installed local system to remote target with hostname dash-phx on the popular Postman tool.

2 Sample classes for Input, Output & Error

Note: The example classes follow C# class style declaration.

2.1 Input class to generate JSON input

The input class is a common class that can be used to create any DASH Command in Developer Mode in DASH CLI as well as to perform REST API communication with AMD Manageability API. It has the fields required to create the input JSON.

Input Class	`public class CommandOptions {`
	`  public string h { get; set; }`
	`  public string u { get; set; }`
	`  public string P { get; set; }`
	`  public List<string> S { get; set; }`
	`  public string a { get; set; }`
	`  public string H { get; set; }`
	`  public string s { get; set; }`
	`  public string e { get; set; }`
	`  public List<string> p { get; set; }`
	`  public int C { get; set; }`
	`  public string t { get; set; }`
	`  public List<string> Commands { get; set; }`
	`  public string cert { get; set; }`
	`  public string key { get; set; }`
	`  public string cacert { get; set; }`
	`  public string r { get; set; }`
	`  public string rp { get; set; }`
	`  public string rc { get; set; }`
	`  public string rk { get; set; }`
	`  public string rca { get; set; }`
	`  public string ssh { get; set; }`
	`  public string conf { get; set; }`
	`}`

The property fields are as follows:

- h – Host Name of DASH system
- u – Username
- P – Password
- S – Transport Scheme (HTTP/HTTPS)
- A – Authentication Mechanism (digest/cert/gss)
- H – Host Array (used only with Discovery)
- s – Start Address (used only with Discovery)
- e – End Address (used only with Discovery)
- p – Port Number
- C – Ignore Certificates (1=> Ignore certificate validation, 0=> Perform certificate validation)
- t – Target Command
- Commands – Commands passed along with the target
- cert – Certificate (used for Mutual TLS use-case)
- key – Certificate Key (used for certificate authentication)
- cacert – CA certificate (used for certificate authentication when ca cert is needed by the server)
- r – ACMS cloud IP
- rp – ACMS cloud port
- rc – ACMS cloud certificate
- rk – ACMS cloud certificate key
- rca – ACMS cloud CA certificate
- ssh – Path to KVM SSH Key

- conf – Path to CertAuthConfi.json file needed for Certificate Authentication

The “Input JSON” inputs can be sent by using the class Command Options to send input to AMA. Each operation has its own class associated, to de-serialize the output from AMA in JSON format.

With AMD Manageability API a user can try the same JSON input via Postman or Curl tool to see the response JSON in Section 0.

2.2 Output class used for different commands

Output Classes for all profiles have a similar format. There is a Profile Collection class, with a List of Profile Members. The Profile Members differ from one profile to the next.

Output Class	

```
public class ProfileCollection {
    [JsonProperty(“@odata.id”)]
    public string _odata_id { get; set; }
    [JsonProperty(“@odata.type”)]
    public string _odata_type { get; set; }
    [JsonProperty(“Members”)]
    public List<ProfileMember> Members { get; set; }
    public string Id { get; set; }
    public string Name { get; set; }
}
public class ProfileMember {
    [JsonProperty(“@odata.id”)]
    public string _odata_id { get; set; }
    [JsonProperty(“@odata.type”)]
    public string _odata_type { get; set; }
    public string Name { get; set; }
    public int InstanceNumber { get; set; }
    … /* Other Properties specific to the profile */
 }
```

2.3 Error class

1) If Error code is Zero, it means no errors have occurred, then the error message is status of executed command.
2) If Error code is non-zero, it means some errors have occurred, then the error message prints the output of the errors for executing the command.

Error Class	

```
public class CommandError {
    public int ErrorCode { get; set; }
    public string ErrorMessage { get; set; }
}
```

For the specific error code, you can find a more detailed description in the section Appendix B.

3 Check Connection and Version

Purpose: This API verifies connectivity and authentication to a DASH-enabled target system.

3.1 Check DASH System Connection

This API is used to validate the DASH username and password for the given system.

DASH CLI Command	**N/A**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":" dash-system","u":"admin","P":"adminpass","C":1,"Commands":["checkconnection"]}**
Output JSON	**{ "Members": [{ "CanConnect": true }] }**
Output Class	`public class CheckConnectionCollection { [JsonProperty("Members")] public List<CheckConnectionMember> Members; } public class CheckConnectionMember { public string CanConnect { get; set; } }`

3.2 Check AMA Version

DASH CLI Command	**dashcli version**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":" dash-system","u":"admin","P":"adminpass","C":1,"Commands":["version"]}**
Output JSON	**{ "Members": [{ "version": "9.0.0.603" }] }**
Output Class	`public class VersionCollection { [JsonProperty("Members")] public List<VersionMember> Members; } public class VersionMember { public string Version { get; set; } }`

4 Discover

Purpose: The discover command is used to discover systems in a network and find out DASH Support information from the DASH system.

4.1 Discover Host

This API discovers a DASH-enabled system using a specified hostname.

DASH CLI Command	`dashcli -h dash-system discover`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","Commands":["discover"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/DiscoveryCollection", "@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection", "Members" : [{ "@odata.id" : "/DASH/v1/DiscoveryCollection/Discover", "@odata.type" : "#Discover.v1_0_0.Discover", "DashPort" : 623, "Discovered" : true, "HostName" : "dash-system", "Name" : "Discover" }], "Name" : "Discovery Collection" }`
Output Class	`public class DiscoveryCollection { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } [JsonProperty("name")] public string name { get; set; } [JsonProperty("Members")] public List<DiscoveryMember> Members; } public class DiscoveryMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } public string HostName { get; set; } public bool Discovered { get; set; } public int DashPort { get; set; } [JsonProperty("Info")] public DiscoveryInfo Info { get; set; } } public class DiscoveryInfo { public string DASHVersion { get; set; } public string ProductVendor { get; set; } public string ProductVersion { get; set; } public string ProtocolVersion { get; set; } public List<string> SecurityProfile { get; set; } }`

4.2 Discover Host with mutual cert

This API discovers a DASH-enabled system using a hostname with mutual certificate authentication.

DASH CLI Command	`dashcli -h dash-system -cert "C:\Program Files\AMD\Manageability API\certs\MutualAuthCert.pem" discover`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","cert":"C:\Program Files\AMD\Manageability API\certs\MutualAuthCert.pem","Commands":["discover"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/DiscoveryCollection",
	"@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/DiscoveryCollection/Discover",
	"@odata.type" : "#Discover.v1_0_0.Discover",
	"DashPort" : 623,
	"Discovered" : true,
	"HostName" : "dash-system",
	"Name" : "Discover"
	}
	],
	"Name" : "Discovery Collection"
	}

Output Class	public class DiscoveryCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	[JsonProperty("name")]
	public string name { get; set; }
	[JsonProperty("Members")]
	public List<DiscoveryMember> Members;
	}
	public class DiscoveryMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string HostName { get; set; }
	public bool Discovered { get; set; }
	public int DashPort { get; set; }
	[JsonProperty("Info")]
	public DiscoveryInfo Info { get; set; }
	}
	public class DiscoveryInfo {
	public string DASHVersion { get; set; }
	public string ProductVendor { get; set; }
	public string ProductVersion { get; set; }
	public string ProtocolVersion { get; set; }
	public List<string> SecurityProfile { get; set; }
	}

4.3 Discover IP

This API discovers a DASH-enabled system using a specified IP address.

DASH CLI Command	`dashcli -h 10.138.137.71 discover`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"10.138.137.71","Commands":["discover"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/DiscoveryCollection",
	"@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/DiscoveryCollection/Discover",
	"@odata.type" : "#Discover.v1_0_0.Discover",
	"DashPort" : 623,
	"Discovered" : true,
	"HostName" : "10.138.137.71",
	"Name" : "Discover"
	}
	],
	"Name" : "Discovery Collection"
	}

Output Class	public class DiscoveryCollection { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")]

	public string _odata_type { get; set; }
	[JsonProperty("name")]
	public string name { get; set; }
	[JsonProperty("Members")]
	public List<DiscoveryMember> Members;
	}
	public class DiscoveryMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string HostName { get; set; }
	public bool Discovered { get; set; }
	public int DashPort { get; set; }
	[JsonProperty("Info")]
	public DiscoveryInfo Info { get; set; }
	}
	public class DiscoveryInfo {
	public string DASHVersion { get; set; }
	public string ProductVendor { get; set; }
	public string ProductVersion { get; set; }
	public string ProtocolVersion { get; set; }
	public List<string> SecurityProfile { get; set; }
	}

4.4 Discover Host Array

This API discovers multiple DASH-enabled systems using a list of hostnames.

DASH CLI Command	**N/A**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"H":["non-dash-system1","dash-system1","non-dash-system2","dash-system2","non-dash-system3","non-dash-system4"], "Commands":["discover"]}**

Output JSON	**{
	"@odata.id" : "/DASH/v1/DiscoveryCollection",
	"@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection",
	"Members" : [
	{
	"Discovered" : false,
	"HostName" : "non-dash-system1"
	},
	{
	"@odata.id" : "/DASH/v1/DiscoveryCollection/Discover",
	"@odata.type" : "#Discover.v1_0_0.Discover",
	"DashPort" : 623,
	"Discovered" : true,
	"HostName" : "dash-system1",
	"Name" : "Discover"
	},
	{
	"Discovered" : false,
	"HostName" : "non-dash-system2"
	},
	{
	"@odata.id" : "/DASH/v1/DiscoveryCollection/Discover",
	"@odata.type" : "#Discover.v1_0_0.Discover",
	"DashPort" : 623,
	"Discovered" : true,
	"HostName" : "dash-system2",
	"Name" : "Discover"
	},
	{
	"Discovered" : false,
	"HostName" : "non-dash-system3"
	},
	{
	"Discovered" : false,
	"HostName" : "non-dash-system4"
	}**

	``
	    ],
	    "Name" : "Discovery Collection"
	}
	``

Output Class	```
	    public class DiscoveryInfo {
	        public string DASHVersion { get; set; }
	        public string ProductVendor { get; set; }
	        public string ProductVersion { get; set; }
	        public string ProtocolVersion { get; set; }
	        public List<string> SecurityProfile { get; set; }
	    }
	    public class DiscoveryMember {
	        [JsonProperty("@odata.id")]
	        public string _odata_id { get; set; }
	        [JsonProperty("@odata.type")]
	        public string _odata_type { get; set; }
	        public string HostName { get; set; }
	        public bool Discovered { get; set; }
	        public int DashPort { get; set; }
	        [JsonProperty("Info")]
	        public DiscoveryInfo Info { get; set; }
	    }
	    public class DiscoveryCollection {
	        [JsonProperty("@odata.id")]
	        public string _odata_id { get; set; }
	        [JsonProperty("@odata.type")]
	        public string _odata_type { get; set; }
	        [JsonProperty("name")]
	        public string name { get; set; }
	        [JsonProperty("Members")]
	        public List<DiscoveryMember> Members;
	    }
	``

4.5 Discover IP Range

This API discovers DASH-enabled systems within a specified IP address range.

DASH CLI Command	`dashcli -s 10.138.135.208 -e 10.138.135.215 discover`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"s":"10.138.135.208","e":"10.138.135.215","Commands":["discover"]}`

Output JSON	```
	{
	   "@odata.id" : "/DASH/v1/DiscoveryCollection",
	   "@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection",
	   "Members" : [
	      {
	         "Discovered" : false,
	         "HostName" : "10.138.135.208"
	      },
	      {
	         "Discovered" : false,
	         "HostName" : "10.138.135.209"
	      },
	      {
	         "Discovered" : false,
	         "HostName" : "10.138.135.210"
	      },
	      {
	         "Discovered" : false,
	         "HostName" : "10.138.135.211"
	      },
	      {
	         "Discovered" : false,
	         "HostName" : "10.138.135.212"
	      },
	      {
	         "Discovered" : false,
	         "HostName" : "10.138.135.213"
	      },
	      {
	         "@odata.id" : "/DASH/v1/DiscoveryCollection/Discover",
	``

	"@odata.type" : "#Discover.v1_0_0.Discover",
	"DashPort" : 623,
	"Discovered" : true,
	"HostName" : "10.138.135.214",
	"Name" : "Discover"
	},
	{
	"Discovered" : false,
	"HostName" : "10.138.135.215"
	}
	],
	"Name" : "Discovery Collection"
	}

Output Class	public class DiscoveryCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	[JsonProperty("name")]
	public string name { get; set; }
	[JsonProperty("Members")]
	public List<DiscoveryMember> Members;
	}
	public class DiscoveryMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string HostName { get; set; }
	public bool Discovered { get; set; }
	public int DashPort { get; set; }
	[JsonProperty("Info")]
	public DiscoveryInfo Info { get; set; }
	}
	public class DiscoveryInfo {
	public string DASHVersion { get; set; }
	public string ProductVendor { get; set; }
	public string ProductVersion { get; set; }
	public string ProtocolVersion { get; set; }
	public List<string> SecurityProfile { get; set; }
	}

4.6 Discover Info

This API retrieves discovery information for a DASH-enabled target system. The discover info command provides some additional information such as Product Vendor, Product Version, Protocol Version, Security Profiles which can be used to determine the authentication and DASH profile support on the system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass discover info`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass",`

	`"Commands":["discover","info"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/DiscoveryCollection", "@odata.type" : "#DiscoveryCollection.v1_0_0.DiscoveryCollection", "Members" : [{ "@odata.id" : "/DASH/v1/DiscoveryCollection/Discover", "@odata.type" : "#Discover.v1_0_0.Discover", "DashPort" : 623, "Discovered" : true, "HostName" : "dash-system", "Info" : { "DASHVersion" : "1.1.0", "ProductVendor" : "Broadcom Corporation", "ProductVersion" : "1.40.0.1 Feb 19 2010 19:54:12", "ProtocolVersion" : "http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd", "SecurityProfile" : ["http://schemas.dmtf.org/wbem/wsman/1/wsman/secprofile/http/digest", "http://schemas.dmtf.org/wbem/wsman/1/wsman/secprofile/http/spnego-kerberos", "http://schemas.dmtf.org/wbem/wsman/1/wsman/secprofile/https/digest", "http://schemas.dmtf.org/wbem/wsman/1/wsman/secprofile/https/spnego-kerberos"] }, "Name" : "Discover" }], "Name" : "Discovery Collection" }`
Output Class	`public class DiscoveryCollection { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } [JsonProperty("name")] public string name { get; set; } [JsonProperty("Members")] public List<DiscoveryMember> Members; } public class DiscoveryMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } public string HostName { get; set; } public bool Discovered { get; set; } public int DashPort { get; set; } [JsonProperty("Info")] public DiscoveryInfo Info { get; set; } } public class DiscoveryInfo { public string DASHVersion { get; set; } public string ProductVendor { get; set; } public string ProductVersion { get; set; } public string ProtocolVersion { get; set; } public List<string> SecurityProfile { get; set; } }`

5 DASH Profiles

DASH specification contains over 40 profiles for Remote System Management.
Sections 6 onwards, we will be discussing each command supported by AMA. At a minimum each command will support enumeration and show operations.
While enumerate profile shows all the instances of the profile command, show command can be used to show the value of any specific instance. To avoid redundancy, we will be discussing enumerate command here and avoid adding enumerate command for each profile command as the data is repetitive.
To help understand this better section 5.1 and 5.2 have a template depiction of enumerate profile and profile show which can be used with any profile.

5.1 Enumerate Profile

This API retrieves a list of all available profile instances on the target system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass enumerate profile`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass", "Commands":["enumerate"," profile"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/ProfileCollection", "@odata.type" : "#ProfileCollection.v1_0_0.ProfileCollection", "Id" : "ProfileCollection", "Members" : [ { "@odata.id" : "/DASH/v1/ProfileCollection/Profile", "@odata.type" : "#Profile.v1_0_0.Profile", … } ], "Name" : "Profile Collection" }`
Output Class	`public class ProfileCollection { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string Id { get; set; } public string _odata_type { get; set; } [JsonProperty("Members")] public List<ProfileMember> Members { get; set; } public string Name { get; set; } } public class ProfileMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } … }`

5.2 Profile Show

This API retrieves detailed information for a specific ACMS instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t profile[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"profile[0]", "Commands":["show"]}`

Output JSON	<pre>{
	  "@odata.id" : "/DASH/v1/ProfileCollection/Profile",
	  "@odata.type" : "#Profile.v1_0_0.Profile",
	  …
	}</pre>

Output Class	<pre>public class ProfileMember {
	  [JsonProperty("@odata.id")]
	  public string _odata_id { get; set; }
	  [JsonProperty("@odata.type")]
	  public string _odata_type { get; set; }
	  …
	}</pre>

5.3 Supported profiles

The following profile commands support this pattern

	Profile	Enumerate Profile
1	acms	"Commands"=["enumerate", "acms"]
2	alertdestination	"Commands"=["enumerate", "alertdestination"]
3	asset	"Commands"=["enumerate", "asset"]
4	battery	"Commands"=["enumerate", "battery"]
5	bios	"Commands"=["enumerate", "bios"]
6	bootconfig	"Commands"=["enumerate", "bootconfig"]
7	computersystem	"Commands"=["enumerate", "computersystem"]
8	dhcpclient	"Commands"=["enumerate", "dhcpclient"]
9	diskinfo	"Commands"=["enumerate", "diskinfo"]
10	dnsclient	"Commands"=["enumerate", "dnsclient"]
11	ethernetport	"Commands"=["enumerate", "ethernetport"]
12	fan	"Commands"=["enumerate", "fan"]
13	filtercollection	"Commands"=["enumerate", "filtercollection"]
14	indicationfilter	"Commands"=["enumerate", "indicationfilter"]
15	indicationsubscription	"Commands"=["enumerate", "indicationsubscription"]
16	indicatorled	"Commands"=["enumerate", "indicatorled"]
17	ipconfiguration	"Commands"=["enumerate", "ipconfiguration"]
18	ipinterface	"Commands"=["enumerate", "ipinterface"]
19	kvmredirection	"Commands"=["enumerate", "kvmredirection"]
20	logentry	"Commands"=["enumerate", "logentry"]
21	mediaredirection	"Commands"=["enumerate", "mediaredirection"]
22	memory	"Commands"=["enumerate", "memory"]
23	networkport	"Commands"=["enumerate", "networkport"]
24	opaquemanagementdata	"Commands"=["enumerate", "opaquemanagementdata"]
25	operatingsystem	"Commands"=["enumerate", "operatingsystem"]
26	pcidevice	"Commands"=["enumerate", "pcidevice"]
27	physicalcomputersystemview	"Commands"=["enumerate", "physicalcomputersystemview"]
28	platformwatchdog	"Commands"=["enumerate", "platformwatchdog"]
29	powerscheduler	"Commands"=["enumerate", "powerscheduler"]
30	powersupply	"Commands"=["enumerate", "powersupply"]

31	processor	`"Commands"=["enumerate", "processor"]`
32	recordlog	`"Commands"=["enumerate", "recordlog"]`
33	registeredprofile	`"Commands"=["enumerate", "registeredprofile"]`
34	role	`"Commands"=["enumerate", "role"]`
35	sensor	`"Commands"=["enumerate", "sensor"]`
36	serviceprocessor	`"Commands"=["enumerate", "serviceprocessor"]`
37	software	`"Commands"=["enumerate", "software"]`
38	softwareinstallationservice	`"Commands"=["enumerate", "softwareinstallationservice"]`
39	softwareinstallationservicecapabilities	`"Commands"=["enumerate", "softwareinstallationservicecapabilities"]`
40	ssh	`"Commands"=["enumerate", "ssh"]`
41	telnet	`"Commands"=["enumerate", "telnet"]`
42	textredirection	`"Commands"=["enumerate", "textredirection"]`
43	upgradablesoftware	`"Commands"=["enumerate", "upgradablesoftware"]`
44	usbredirection	`"Commands"=["enumerate", "usbredirection"]`
45	user	`"Commands"=["enumerate", "user"]`
46	wifiport	`"Commands"=["enumerate", "wifiport"]`

5.4 Enumerate All

Purpose: Enumerate all the profiles.

To ease the job of console testing for all the profiles, AMA has a special command that allows you to enumerate all the profile commands.

Lists output of all the profiles and their instances on the target system in one command to give a comprehensive output of DASH related information.

DASH CLI Command	`N/A`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","Commands":["enumerate", "all"]}`

The output of this command is a large JSON that contains the results of all the individual profile commands enumeration discussed in this document.

6 ACMS

Purpose: The AMD Cloud Manageability Service (ACMS) configuration settings on an AIM-T enabled system can be retrieved.

6.1 ACMS Show

This API retrieves detailed information for a specific ACMS instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t acms[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"acms[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/ACMSCollection/ACMS", "@odata.type" : "#ACMS.v1_0_0.ACMS", "CreationClassName" : "CIM_ListenerDestinationWSManagement", "Destination" : "http://192.168.56.1:3275/eventsink", "ElementName" : "WS-Management Listener Destination #1", "InstanceNumber" : 0, "Name" : "ListenerDestinationWSManagement:1", "PersistenceType" : "Transient", "Protocol" : "WS-Management", "SystemCreationClassName" : "CIM_ComputerSystem", "SystemName" : "dash-system" }`
Output Class	`public class ACMSMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } public string CreationClassName { get; set; } public string Destination { get; set; } public string ElementName { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string PersistenceType { get; set; } public string Protocol { get; set; } public string SystemCreationClassName { get; set; } public string SystemName { get; set; } }`

7 Alert Destination

Purpose: Management of alert/event notification destinations.

7.1 Alert Destination Show

This API retrieves detailed information for a specific alert destination.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t alertdestination[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"alertdestination[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/AlertDestinationCollection/AlertDestination", "@odata.type" : "#AlertDestination.v1_0_0.AlertDestination", "CreationClassName" : "CIM_ListenerDestinationWSManagement", "Destination" : "http://192.168.56.1:3275/eventsink", "ElementName" : "WS-Management Listener Destination #1", "InstanceNumber" : 0, "Name" : "ListenerDestinationWSManagement:1", "PersistenceType" : "Transient", "Protocol" : "WS-Management", "SystemCreationClassName" : "CIM_ComputerSystem", "SystemName" : "dash-system" }`
Output Class	`public class AlertDestinationMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } public string CreationClassName { get; set; } public string Destination { get; set; } public string ElementName { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string PersistenceType { get; set; } public string Protocol { get; set; } public string SystemCreationClassName { get; set; } public string SystemName { get; set; } }`

7.2 Alert Destination Delete

This API deletes a specified alert destination from the target system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t alertdestination[0] delete`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"alertdestination[0]", "Commands":["delete"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Alert Destination Deleted Successfully" }`
Output Class	`public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

8 Asset

Purpose: Physical Hardware asset inventory management.

8.1 Asset Show

This API retrieves detailed asset information for a specific asset instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t asset[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"asset[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset", "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset", "ConnectorLayout" : "Slot", "ConnectorType" : ["PCI"], "ElementName" : "Slot #1", "InstanceNumber" : 0, "Manufacturer" : "N/A", "Model" : "N/A", "Name" : "PhysicalAsset", "PartNumber" : "N/A", "SKU" : "N/A", "SerialNumber" : "N/A", "SlotNumber" : 4, "Tag" : "BRCM:6.1" }`
Output Class	`public class PhysicalAssetMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string ConnectorLayout { get; set; } public List<string> ConnectorType { get; set; } public string ElementName { get; set; } public int InstanceNumber { get; set; } public string Manufacturer { get; set; } public string Model { get; set; } public string Name { get; set; } public string PartNumber { get; set; } public string SKU { get; set; } public string SerialNumber { get; set; } public int SlotNumber { get; set; } public string Tag { get; set; } public string CanbeFRUed { get; set; } public string MemoryBankLabel { get; set; } public string MemoryCapacity { get; set; } public string MemoryFormFactor { get; set; } public string MemoryType { get; set; } public string IsHostingBoard { get; set; } public string PackageType { get; set; } public string ChassisType { get; set; } }`

9 Battery

Purpose: Battery status and composition can be read.

9.1 Battery Show

This API retrieves detailed information for a specific battery instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t battery[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"battery[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/BatteryCollection/Battery", "@odata.type" : "#Battery.v1_0_0.Battery", "BatteryStatus" : "Fully Charged", "Chemistry" : "N/A", "DeviceID" : "Battery:0", "ElementName" : "Battery:0", "EnabledState" : "Enabled", "EstimatedRunTime" : "N/A", "ExpectedLife" : "N/A", "HealthState" : "OK", "InstanceNumber" : 0, "MaxRechargeCount" : "N/A", "MaxRechargeTime" : "N/A", "Name" : "Battery", "OperationalState" : "N/A", "RechargeCount" : "N/A", "RequestedState" : "Enabled", "TimeToFullRecharge" : "N/A" }`

Output Class	`public class Battery { public string odataid { get; set; } public string odatatype { get; set; } public string Id { get; set; } public BatteryMember[] Members { get; set; } public string Name { get; set; } }`
	
	`public class BatteryMember { public string odataid { get; set; } public string odatatype { get; set; } public string BatteryStatus { get; set; } public string Chemistry { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string EstimatedRunTime { get; set; } public string ExpectedLife { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string MaxRechargeCount { get; set; } public string MaxRechargeTime { get; set; } public string Name { get; set; } public string OperationalState { get; set; } public string RechargeCount { get; set; } public string RequestedState { get; set; } public string TimeToFullRecharge { get; set; } }`

	}

9.2 Battery Enable

This API enables the specified battery instance on the target system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t battery[0] enable`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"battery[0]", "Commands":["enable"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Battery Tested successfully", }`
Output Class	`public class BatteryEnableStatus { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

9.3 Battery Disable

This API disables the specified battery instance on the target system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t battery[0] disable`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"battery[0]", "Commands":["disable"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Battery Disabled successfully", }`
Output Class	`public class BatteryDisableStatus { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

9.4 Battery Test

This API initiates a self-test operation for the specified battery.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t battery[0] test`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"battery[0]", "Commands":["test"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Battery Tested successfully", }`
Output Class	`public class BatteryTestStatus { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

9.5 Battery Recharge

This API initiates a recharge operation for the specified battery.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t battery[0] recharge`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`

Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"battery[0]", "Commands":["recharge"]}

Output JSON	{
	"ErrorCode" : 0,
	"ErrorMessage" : "Battery Recharged successfully",
	}

Output Class	public class BatteryRechargeStatus
	{
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

10 BIOS

Purpose: BIOS configuration and attribute management.

10.1 BIOS Show

This API retrieves detailed BIOS information for the specified instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t bios[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"bios[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/BIOSCollection/BIOS", "@odata.type" : "#BIOS.v1_0_0.BIOS", "BIOSAttributes" : [{ "AllowedStringLength" : "N/A", "AttributeInstanceNumber" : 0, "AttributeName" : "Com Port Address", "CurrentValues" : ["Disabled"], "DefaultValues" : ["Disabled"], "InstanceID" : "BIOSEnumeration:2147680256", "IsOrderedList" : "N/A", "IsReadOnly" : "False", "PendingValues" : "N/A", "PossibleValues" : ["Disabled", "Enabled"], "StringType" : "N/A" }, { "AllowedStringLength" : "N/A", "AttributeInstanceNumber" : 1, "AttributeName" : "Route Port 80h Cycles to", "CurrentValues" : ["LPC"], "DefaultValues" : ["LPC"], "InstanceID" : "BIOSEnumeration:2147680257", "IsOrderedList" : "N/A", "IsReadOnly" : "False", "PendingValues" : "N/A", "PossibleValues" : ["LPC", "PCI"], "StringType" : "N/A" }, { "AllowedStringLength" : "N/A", "AttributeInstanceNumber" : 2, "AttributeName" : "Legacy USB Support", "CurrentValues" : ["Enabled"], "DefaultValues" : ["Disabled"], "InstanceID" : "BIOSEnumeration:2147680258", "IsOrderedList" : "N/A", "IsReadOnly" : "False", "PendingValues" : "N/A", "PossibleValues" : ["Disabled", "Enabled"], "StringType" : "N/A" }, { "AllowedStringLength" : "N/A", "AttributeInstanceNumber" : 3, "AttributeName" : "Force PXE Boot", "CurrentValues" : ["Enabled"], "DefaultValues" : ["Disabled"], "InstanceID" : "BIOSEnumeration:2147680259", "IsOrderedList" : "N/A", "IsReadOnly" : "False", "PendingValues" : "N/A", "PossibleValues" : ["Disabled", "Enabled"], "StringType" : "N/A" },`

<table>
<tr><td></td><td>


```
        {
           "AllowedStringLength" : "N/A",
           "AttributeInstanceNumber" : 4,
           "AttributeName" : "Force HardDisk0 Boot",
           "CurrentValues" : [ "Enabled" ],
           "DefaultValues" : [ "Disabled" ],
           "InstanceID" : "BIOSEnumeration:2147680260",
           "IsOrderedList" : "N/A",
           "IsReadOnly" : "False",
           "PendingValues" : "N/A",
           "PossibleValues" : [ "Disabled", "Enabled" ],
           "StringType" : "N/A"
        },
        {
           "AllowedStringLength" : "N/A",
           "AttributeInstanceNumber" : 5,
           "AttributeName" : "Force HardDisk1 Safe Boot",
           "CurrentValues" : [ "Enabled" ],
           "DefaultValues" : [ "Disabled" ],
           "InstanceID" : "BIOSEnumeration:2147680261",
           "IsOrderedList" : "N/A",
           "IsReadOnly" : "False",
           "PendingValues" : "N/A",
           "PossibleValues" : [ "Disabled", "Enabled" ],
           "StringType" : "N/A"
        },
        {
           "AllowedStringLength" : "N/A",
           "AttributeInstanceNumber" : 6,
           "AttributeName" : "Force HardDisk2 Diag",
           "CurrentValues" : [ "Enabled" ],
           "DefaultValues" : [ "Disabled" ],
           "InstanceID" : "BIOSEnumeration:2147680262",
           "IsOrderedList" : "N/A",
           "IsReadOnly" : "False",
           "PendingValues" : "N/A",
           "PossibleValues" : [ "Disabled", "Enabled" ],
           "StringType" : "N/A"
        },
        {
           "AllowedStringLength" : "N/A",
           "AttributeInstanceNumber" : 7,
           "AttributeName" : "Force CD/DVD Rom Boot",
           "CurrentValues" : [ "Enabled" ],
           "DefaultValues" : [ "Disabled" ],
           "InstanceID" : "BIOSEnumeration:2147680263",
           "IsOrderedList" : "N/A",
           "IsReadOnly" : "False",
           "PendingValues" : "N/A",
           "PossibleValues" : [ "Disabled", "Enabled" ],
           "StringType" : "N/A"
        }
     ],
     "BIOSName" : "10M30009US",
     "InstanceNumber" : 0,
     "IsPrimaryBIOS" : "N/A",
     "Manufacturer" : "LENOVO",
     "Name" : "BIOS",
     "SoftwareElementID" : "BIOS:262148",
     "SoftwareElementState" : "Executable",
     "TargetOperatingSystem" : "Not Applicable",
     "Version" : "M11KT45A"
}
```


</td></tr>
<tr><td>Output Class</td><td>

```
public class BIOSMember {
    [JsonProperty("@odata.id")]
    public string _odata_id { get; set; }
    [JsonProperty("@odata.type")]
    public string odatatype { get; set; }
    public List<BiosAttribute> BIOSAttributes { get; set; }
```

</td></tr>
</table>

	public string BIOSName { get; set; }
	public int InstanceNumber { get; set; }
	public string IsPrimaryBIOS { get; set; }
	public string Manufacturer { get; set; }
	public string Name { get; set; }
	public string SoftwareElementID { get; set; }
	public string SoftwareElementState { get; set; }
	public string TargetOperatingSystem { get; set; }
	public string Version { get; set; }
	}
	public class BiosAttribute {
	public string AllowedStringLength { get; set; }
	public int AttributeInstanceNumber { get; set; }
	public string AttributeName { get; set; }
	public List<string> CurrentValues { get; set; }
	public List<string> DefaultValues { get; set; }
	public string InstanceID { get; set; }
	public string IsOrderedList { get; set; }
	public string IsReadOnly { get; set; }
	public List<string> PendingValues { get; set; }
	public List<string> PossibleValues { get; set; }
	public string StringType { get; set; }
	}

10.2 BIOS Set Attribute

This API updates a specified BIOS attribute on the target system.

DASH CLI Command	**dashcli -h dash-system -u admin -P adminpass -t bios[0] setattribute "Com Port Address" Enable**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":"dash-system","u":"admin","P":"adminpass","t":"bios[0]", "Commands":["setattribute","Com Port Address","Enable"]}**
Output JSON	**{ "ErrorCode" : 0, "ErrorMessage" : "SetAttributes Successful" }**

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

11 Boot Config

Purpose: Boot configuration and boot order management.

11.1 Boot Config Show

This API retrieves detailed boot configuration information for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t bootconfig[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"bootconfig[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/BootConfigurationCollection/BootConfiguration",
	"@odata.type" : "#BootConfiguration.v1_0_0.BootConfiguration",
	"BootDevices" : [
	{
	"Name" : "BRCM:Hard Drive:1",
	"Value" : 0
	},
	{
	"Name" : "BRCM:USB Device:4",
	"Value" : 1
	},
	{
	"Name" : "BRCM:Network:3",
	"Value" : 2
	},
	{
	"Name" : "BRCM:CD-ROM:2",
	"Value" : 3
	}
	],
	"ElementName" : "Boot Configuration Setting #1",
	"Id" : "BootConfiguration",
	"InstanceId" : "BRCM:31.1",
	"InstanceNumber" : 0,
	"IsCurrentConfiguration" : "No",
	"IsDefaultConfiguration" : "No",
	"IsNextConfiguration" : "Yes",
	"Name" : "Boot Configuration"
	}

Output Class	public class BootConfigurationMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public List<BootDevice> BootDevices { get; set; }
	public string ElementName { get; set; }
	public string Id { get; set; }
	public string InstanceId { get; set; }
	public int InstanceNumber { get; set; }
	public string IsCurrentConfiguration { get; set; }
	public string IsDefaultConfiguration { get; set; }
	public string IsNextConfiguration { get; set; }
	public string Name { get; set; }
	}
	public class BootDevice {
	public string Name { get; set; }
	public int Value { get; set; }
	}

11.2 Boot Config Change Boot Order

This API changes/updates the boot order configuration for the target system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t bootconfig[0] changebootorder 3 2 1 0`

jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"bootconfig[0]", "Commands":["changebootorder","3","2","1","0"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Boot Order Changed Successfully" }`
Output Class	`public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

12 Computer System

Purpose: System details Ownership and Power configuration and management.

12.1 Computer System Show

This API retrieves detailed information for a specific Computer System instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/ComputerSystemCollection/ComputerSystem", "@odata.type" : "#ComputerSystem.v1_0_1.ComputerSystem", "AvailableRequestedPowerStates" : ["On", "Power Cycle (Off - Soft)", "Off - Soft", "Master Bus Reset"], "CurrentPowerState" : "On", "DedicatedTo" : ["Desktop"], "ElementName" : "Computer System", "EnabledState" : "Enabled", "Id" : "ComputerSystem", "InstanceNumber" : 0, "Name" : "10.136.6.63", "PowerOnTime" : "N/A", "PrimaryOwner" : "Ganesh Kamath", "PrimaryOwnerContact" : "2B550", "RequestSupportedPowerStates" : ["On", "Power Cycle (Off - Soft)", "Off - Soft", "Master Bus Reset", "Sleep -Deep", "Hibernate (Off - Soft)", "Off - Soft Graceful", "Master Bus Reset Graceful", "Power Cycle (Off - Soft Graceful)"], "RequestedPowerState" : "Master Bus Reset", "RequestedState" : "No Change", "SupportedPowerChangeCapabilities" : ["Power State Settable", "Power Cycling Supported", "HW Reset Supported", "Graceful Shutdown Supported"], "SupportedPowerStates" : ["Other", "On", "Sleep - Light", "Sleep -Deep", "Power Cycle (Off - Soft)", "Hibernate (Off - Soft)", "Off - Soft", "Master Bus Reset", "Off - Soft Graceful", "Master Bus Reset Graceful", "Power Cycle (Off - Soft Graceful)"] }`
Output Class	`public class ComputerSystemMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public List<string> AvailableRequestedPowerStates { get; set; } public string CurrentPowerState { get; set; }`

	public List<string> DedicatedTo { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string Id { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string PowerOnTime { get; set; } public string PrimaryOwner { get; set; } public string PrimaryOwnerContact { get; set; } public List<string> RequestSupportedPowerStates { get; set; } public string RequestedPowerState { get; set; } public string RequestedState { get; set; } public List<string> SupportedPowerChangeCapabilities { get; set; } public List<string> SupportedPowerStates { get; set; } }

12.2 Enumerate Asset

This API retrieves asset information associated with the ComputerSystem resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate asset`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","asset"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/PhysicalAssetCollection",
	"@odata.type" : "#PhysicalAssetCollection.v1_0_0.PhysicalAssetCollection",
	"Id" : "PhysicalAssetCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
	"@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
	"CanbeFRUed" : "No",
	"ChassisType" : "Mini Tower",
	"ElementName" : "Chassis",
	"InstanceNumber" : 0,
	"Manufacturer" : "Hewlett-Packard",
	"Model" : "HP Compaq 6005 Pro MT PC",
	"Name" : "PhysicalAsset",
	"PackageType" : "Chassis/Frame",
	"PartNumber" : "N/A",
	"SKU" : "AT493AV",
	"SerialNumber" : "INA105T9LB",
	"Tag" : "BRCM:3"
	}
	],
	"Name" : "Physical Asset Collection"
	}

Output Class	public class PhysicalAssetCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	[JsonProperty("Members")]
	public List<PhysicalAssetMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }

```
        public int InstanceNumber { get; set; }
        public string Manufacturer { get; set; }
        public string Model { get; set; }
        public string Name { get; set; }
        public string PartNumber { get; set; }
        public string SKU { get; set; }
        public string SerialNumber { get; set; }
        public int SlotNumber { get; set; }
        public string Tag { get; set; }
        public string CanbeFRUed { get; set; }
        public string MemoryBankLabel { get; set; }
        public string MemoryCapacity { get; set; }
        public string MemoryFormFactor { get; set; }
        public string MemoryType { get; set; }
        public string IsHostingBoard { get; set; }
        public string PackageType { get; set; }
        public string ChassisType { get; set; }
    }
```

12.3 Enumerate Boot Config

This API retrieves boot configuration information associated with the ComputerSystem resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate bootconfig`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","bootconfig"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/BootConfigurationCollection", "@odata.type" : "#BootConfigurationCollection.v1_0_0.BootConfigurationCollection", "Id" : "BootConfigurationCollection", "Members" : [{ "@odata.id" : "/DASH/v1/BootConfigurationCollection/BootConfiguration", "@odata.type" : "#BootConfiguration.v1_0_0.BootConfiguration", "BootDevices" : [{ "Name" : "BRCM:Network:3", "Value" : 0 }, { "Name" : "BRCM:Hard Drive:1", "Value" : 1 }, { "Name" : "BRCM:USB Device:4", "Value" : 2 }, { "Name" : "BRCM:CD-ROM:2", "Value" : 3 }], "ElementName" : "Boot Configuration Setting #1", "Id" : "BootConfiguration", "InstanceId" : "BRCM:31.1", "InstanceNumber" : 0, "IsCurrentConfiguration" : "No", "IsDefaultConfiguration" : "No", "IsNextConfiguration" : "Yes", "Name" : "Boot Configuration" }, { "@odata.id" : "/DASH/v1/BootConfigurationCollection/BootConfiguration",`

	"@odata.type" : "#BootConfiguration.v1_0_0.BootConfiguration",
	"BootDevices" : [
	{
	"Name" : "BRCM:Hard Drive:5",
	"Value" : 0
	},
	{
	"Name" : "BRCM:CD-ROM:6",
	"Value" : 1
	},
	{
	"Name" : "BRCM:Network:7",
	"Value" : 2
	},
	{
	"Name" : "BRCM:USB Device:8",
	"Value" : 3
	}
	],
	"ElementName" : "Boot Configuration Setting #2",
	"Id" : "BootConfiguration",
	"InstanceId" : "BRCM:31.2",
	"InstanceNumber" : 1,
	"IsCurrentConfiguration" : "No",
	"IsDefaultConfiguration" : "No",
	"IsNextConfiguration" : "No",
	"Name" : "Boot Configuration"
	}
	],
	"Name" : "Boot Config Collection"
	}

Output Class	public class BootConfigurationCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	[JsonProperty("Members")]
	public List<BootConfigurationMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class BootConfigurationMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public List<BootDevice> BootDevices { get; set; }
	public string ElementName { get; set; }
	public string Id { get; set; }
	public string InstanceId { get; set; }
	public int InstanceNumber { get; set; }
	public string IsCurrentConfiguration { get; set; }
	public string IsDefaultConfiguration { get; set; }
	public string IsNextConfiguration { get; set; }
	public string Name { get; set; }
	}
	public class BootDevice {
	public string Name { get; set; }
	public int Value { get; set; }
	}

12.4 Enumerate Fan

This API retrieves fan information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate fan`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`

Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","fan"]}

Output JSON	{
	"@odata.id" : "/DASH/v1/FanCollection",
	"@odata.type" : "#FanCollection.v1_0_0.FanCollection",
	"Id" : "FanCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/FanCollection/Fan",
	"@odata.type" : "#Fan.v1_0_0.Fan",
	"DesiredSpeed" : "N/A",
	"DeviceID" : "BRCM:10.1",
	"ElementName" : "Fan #1",
	"EnabledState" : "Not Applicable",
	"HealthState" : "OK",
	"InstanceNumber" : 0,
	"IsActiveCooling" : "Yes",
	"IsVariableSpeed" : "Yes",
	"Name" : "Fan",
	"OperationalState" : ["OK"],
	"RequestedState" : "Not Applicable"
	},
	{
	"@odata.id" : "/DASH/v1/FanCollection/Fan",
	"@odata.type" : "#Fan.v1_0_0.Fan",
	"DesiredSpeed" : "N/A",
	"DeviceID" : "BRCM:10.2",
	"ElementName" : "Fan #2",
	"EnabledState" : "Not Applicable",
	"HealthState" : "OK",
	"InstanceNumber" : 1,
	"IsActiveCooling" : "Yes",
	"IsVariableSpeed" : "Yes",
	"Name" : "Fan",
	"OperationalState" : ["OK"],
	"RequestedState" : "Not Applicable"
	}
	],
	"Name" : "Fan Collection"
	}

Output Class	```
	public class FanCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	[JsonProperty("Members")]
	public List<FanMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class FanMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public int DesiredSpeed { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string IsActiveCooling { get; set; }
	public string IsVariableSpeed { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public string RequestedState { get; set; }
	}
	``

12.5 Enumerate Memory

This API retrieves memory information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate memory`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","memory"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/MemoryCollection",
	"@odata.type" : "#MemoryCollection.v1_0_0.MemoryCollection",
	"Id" : "MemoryCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/MemoryCollection/Memory",
	"@odata.type" : "#Memory.v1_0_0.Memory",
	"AccessType" : "Read/Write Supported",
	"AvailableMemory" : "4096 MB",
	"DeviceID" : "BRCM:8.1",
	"ElementName" : "Total System Memory",
	"EnabledState" : "Enabled",
	"HealthState" : "Unknown",
	"InstanceNumber" : 0,
	"IsVolatileMemory" : "Yes",
	"Name" : "Memory",
	"RequestedState" : "Not Applicable",
	"TotalMemory" : "4096 MB"
	}
	],
	"Name" : "Memory Collection"
	}

Output Class	`public class MemoryCollection {`
	`[JsonProperty("@odata.id")]`
	`public string _odata_id { get; set; }`
	`[JsonProperty("@odata.type")]`
	`public string _odata_type { get; set; }`
	`public string Id { get; set; }`
	`[JsonProperty("Members")]`
	`public List<MemoryMember> Members { get; set; }`
	`public string Name { get; set; }`
	`}`
	`public class MemoryMember {`
	`[JsonProperty("@odata.id")]`
	`public string _odata_id { get; set; }`
	`[JsonProperty("@odata.type")]`
	`public string odatatype { get; set; }`
	`public string AccessType { get; set; }`
	`public string AvailableMemory { get; set; }`
	`public string DeviceID { get; set; }`
	`public string ElementName { get; set; }`
	`public string EnabledState { get; set; }`
	`public string HealthState { get; set; }`
	`public int InstanceNumber { get; set; }`
	`public string IsVolatileMemory { get; set; }`
	`public string Name { get; set; }`
	`public string RequestedState { get; set; }`
	`public string TotalMemory { get; set; }`
	`}`

12.6 Enumerate Power Supply

This API retrieves power supply information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate powersupply`
jdo Usage	`dashcli -jdo`

URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate"," powersupply"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PowerSupplyCollection", "@odata.type" : "#PowerSupplyCollection.v1_0_0.PowerSupplyCollection", "Id" : "PowerSupplyCollection", "Members" : [{ "@odata.id" : "/DASH/v1/PowerSupplyCollection/PowerSupply", "@odata.type" : "#PowerSupply.v1_0_0.PowerSupply", "DeviceID" : "BRCM:9.1", "ElementName" : "Power Supply #1", "EnabledState" : "Enabled", "HealthState" : "Unknown", "InstanceNumber" : 0, "Name" : "Power Supply", "OperationalState" : ["Unknown"], "RequestedState" : "Not Applicable", "TotalPower" : 0 }], "Name" : "Power Supply Collection" }`
Output Class	`public class PowerSupplyCollection { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } public string Id { get; set; } [JsonProperty("Members")] public List<PowerSupplyMember> Members { get; set; } public string Name { get; set; } } public class PowerSupplyMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public List<string> OperationalState { get; set; } public string RequestedState { get; set; } public int TotalPower { get; set; } }`

12.7 Enumerate Processor

This API retrieves processor information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate processor`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","processor"]}`
Output JSON	`{ "@odata.id": "/DASH/v1/ProcessorCollection", "@odata.type": "#ProcessorCollection.v1_0_0.ProcessorCollection", "Id": "ProcessorCollection", "Members": [ { "@odata.id": "/DASH/v1/ProcessorCollection/Processor", "@odata.type": "#Processor.v1_0_0.Processor",`

	"CPUStatus": "CPU Enabled", "CurClockSpeed": "2000 MHz", "DeviceID": "AMD:20.0", "ElementName": "Processor:0", "EnabledState": "Enabled", "ExtBusSpeed": "100 MHz", "Family": "AMD Zen(TM) Processor Family", "HealthState": "OK", "InstanceNumber": 0, "LoadPercent": "3%", "MaxClockSpeed": "5150 MHz", "Name": "Processor", "NumberOfCores": 8, "NumberOfEnabledCores": 8, "NumberOfThreadsPerProcessor": 16, "OperationalState": ["OK"], "RequestedState": "Not Applicable", "Stepping": "AMD Ryzen AI 7 PRO " }], "Name": "Processor Collection" }

Output Class	`public class ProcessorCollection`
	`{`
	`public string odataid { get; set; }`
	`public string odatatype { get; set; }`
	`public string Id { get; set; }`
	`public List <ProcessorMember> Members { get; set; }`
	`public string Name { get; set; }`
	`}`
	
	`public class ProcessorMember`
	`{`
	`public string odataid { get; set; }`
	`public string odatatype { get; set; }`
	`public string CPUStatus { get; set; }`
	`public string CurClockSpeed { get; set; }`
	`public string DeviceID { get; set; }`
	`public string ElementName { get; set; }`
	`public string EnabledState { get; set; }`
	`public string ExtBusSpeed { get; set; }`
	`public string Family { get; set; }`
	`public string HealthState { get; set; }`
	`public int InstanceNumber { get; set; }`
	`public string LoadPercent { get; set; }`
	`public string MaxClockSpeed { get; set; }`
	`public string Name { get; set; }`
	`public int NumberOfCores { get; set; }`
	`public int NumberOfEnabledCores { get; set; }`
	`public int NumberOfThreadsPerProcessor { get; set; }`
	`public List <string> OperationalState { get; set; }`
	`public string RequestedState { get; set; }`
	`public string Stepping { get; set; }`
	`}`

12.8 Enumerate Sensor

This API retrieves sensor information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate sensor`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","sensor"]}`

Output JSON


```
{
   "@odata.id" : "/DASH/v1/SensorCollection",
   "@odata.type" : "#SensorCollection.v1_0_0.SensorCollection",
   "Id" : "SensorCollection",
   "Members" : [
      {
         "@odata.id" : "/DASH/v1/SensorCollection/Sensor",
         "@odata.type" : "#Sensor.v1_0_0.Sensor",
         "BaseUnit" : "RPM",
         "CurrentState" : "Normal",
         "DeviceID" : "BRCM:11.1",
         "ElementName" : "Numeric Sensor #1 (Tachometer)",
         "EnabledState" : "Not Applicable",
         "HealthState" : "Unknown",
         "InstanceNumber" : 0,
         "Name" : "Sensor",
         "OperationalState" : [ "No Contact" ],
         "PossibleStates" : [
            "Non-Critical",
            "Lower Non-Critical",
            "Upper Non-Critical",
            "Critical",
            "Lower Critical",
            "Upper Critical",
            "Fatal",
            "Lower Fatal",
            "Upper Fatal",
            "Normal",
            "Unknown"
         ],
         "RateUnits" : "None",
         "RequestedState" : "No Change",
         "SensorReading" : 0,
         "SensorType" : "Tachometer",
         "ThresholdLC" : "N/A",
         "ThresholdLF" : "N/A",
         "ThresholdLNC" : "N/A",
         "ThresholdUC" : "N/A",
         "ThresholdUF" : "N/A",
         "ThresholdUNC" : "N/A",
         "UnitModifier" : 0
      },
      {
         "@odata.id" : "/DASH/v1/SensorCollection/Sensor",
         "@odata.type" : "#Sensor.v1_0_0.Sensor",
         "BaseUnit" : "RPM",
         "CurrentState" : "Normal",
         "DeviceID" : "BRCM:11.2",
         "ElementName" : "Numeric Sensor #2 (Tachometer)",
         "EnabledState" : "Not Applicable",
         "HealthState" : "Unknown",
         "InstanceNumber" : 1,
         "Name" : "Sensor",
         "OperationalState" : [ "No Contact" ],
         "PossibleStates" : [
            "Non-Critical",
            "Lower Non-Critical",
            "Upper Non-Critical",
            "Critical",
            "Lower Critical",
            "Upper Critical",
            "Fatal",
            "Lower Fatal",
            "Upper Fatal",
            "Normal",
            "Unknown"
         ],
         "RateUnits" : "None",
         "RequestedState" : "No Change",
```

```
            "SensorReading" : 0,
            "SensorType" : "Tachometer",
            "ThresholdLC" : "N/A",
            "ThresholdLF" : "N/A",
            "ThresholdLNC" : "N/A",
            "ThresholdUC" : "N/A",
            "ThresholdUF" : "N/A",
            "ThresholdUNC" : "N/A",
            "UnitModifier" : 0
         },
         {
            "@odata.id" : "/DASH/v1/SensorCollection/Sensor",
            "@odata.type" : "#Sensor.v1_0_0.Sensor",
            "BaseUnit" : "Degrees C",
            "CurrentState" : "Normal",
            "DeviceID" : "BRCM:11.3",
            "ElementName" : "Numeric Sensor #3 (Chassis Temperature)",
            "EnabledState" : "Not Applicable",
            "HealthState" : "Unknown",
            "InstanceNumber" : 2,
            "Name" : "Sensor",
            "OperationalState" : [ "No Contact" ],
            "PossibleStates" : [
               "Non-Critical",
               "Lower Non-Critical",
               "Upper Non-Critical",
               "Critical",
               "Lower Critical",
               "Upper Critical",
               "Fatal",
               "Lower Fatal",
               "Upper Fatal",
               "Normal",
               "Unknown"
            ],
            "RateUnits" : "None",
            "RequestedState" : "No Change",
            "SensorReading" : 0,
            "SensorType" : "Temperature",
            "ThresholdLC" : "N/A",
            "ThresholdLF" : "N/A",
            "ThresholdLNC" : "N/A",
            "ThresholdUC" : "N/A",
            "ThresholdUF" : "N/A",
            "ThresholdUNC" : "N/A",
            "UnitModifier" : -3
         },
         {
            "@odata.id" : "/DASH/v1/SensorCollection/Sensor",
            "@odata.type" : "#Sensor.v1_0_0.Sensor",
            "BaseUnit" : "Degrees C",
            "CurrentState" : "Normal",
            "DeviceID" : "BRCM:11.4",
            "ElementName" : "Numeric Sensor #4 (CPU Temperature)",
            "EnabledState" : "Not Applicable",
            "HealthState" : "Unknown",
            "InstanceNumber" : 3,
            "Name" : "Sensor",
            "OperationalState" : [ "No Contact" ],
            "PossibleStates" : [
               "Non-Critical",
               "Lower Non-Critical",
               "Upper Non-Critical",
               "Critical",
               "Lower Critical",
               "Upper Critical",
               "Fatal",
               "Lower Fatal",
               "Upper Fatal",
```

<table>
<tr><td></td><td><pre>
            "Normal",
            "Unknown"
         ],
         "RateUnits" : "None",
         "RequestedState" : "No Change",
         "SensorReading" : 0,
         "SensorType" : "Temperature",
         "ThresholdLC" : "N/A",
         "ThresholdLF" : "N/A",
         "ThresholdLNC" : "N/A",
         "ThresholdUC" : "N/A",
         "ThresholdUF" : "N/A",
         "ThresholdUNC" : "N/A",
         "UnitModifier" : -3
      }
   ],
   "Name" : "Sensor Collection"
}
</pre></td></tr>
<tr><td>Output Class</td><td><pre>
public class SensorCollection {
    [JsonProperty("@odata.id")]
    public string _odata_id { get; set; }
    [JsonProperty("@odata.type")]
    public string _odata_type { get; set; }
    public string Id { get; set; }
    [JsonProperty("Members")]
    public List<SensorMember> Members { get; set; }
    public string Name { get; set; }
}
public class SensorMember {
    [JsonProperty("@odata.id")]
    public string _odata_id { get; set; }
    [JsonProperty("@odata.type")]
    public string odatatype { get; set; }
    public string BaseUnit { get; set; }
    public string CurrentState { get; set; }
    public string DeviceID { get; set; }
    public string ElementName { get; set; }
    public string EnabledState { get; set; }
    public string HealthState { get; set; }
    public int InstanceNumber { get; set; }
    public string Name { get; set; }
    public List<string> OperationalState { get; set; }
    public List<string> PossibleStates { get; set; }
    public string RateUnits { get; set; }
    public string RequestedState { get; set; }
    public int SensorReading { get; set; }
    public string SensorType { get; set; }
    public int ThresholdLC { get; set; }
    public int ThresholdLF { get; set; }
    public int ThresholdLNC { get; set; }
    public int ThresholdUC { get; set; }
    public int ThresholdUF { get; set; }
    public int ThresholdUNC { get; set; }
    public int UnitModifier { get; set; }
}
</pre></td></tr>
</table>

12.9 Enumerate Software

This API retrieves software inventory information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate software`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","software"]}`

Output JSON


```
{
   "@odata.id" : "/DASH/v1/SoftwareCollection",
   "@odata.type" : "#SoftwareCollection.v1_0_0.SoftwareCollection",
   "Id" : "SoftwareCollection",
   "Members" : [
      {
         "@odata.id" : "/DASH/v1/SoftwareCollection/Software",
         "@odata.type" : "#Software.v1_0_0.Software",
         "BuildNumber" : 1,
         "ElementName" : "Bios Firmware",
         "IdentityInfoType" : [ "Bios Firmware" ],
         "IdentityInfoValue" : [ "BIOS:285409537:0" ],
         "InstanceID" : "SW:0",
         "InstanceNumber" : 0,
         "IsEntity" : "True",
         "MajorVersion" : 1,
         "Manufacturer" : "LENOVO",
         "MinorVersion" : 0,
         "Name" : "Software",
         "OperationalStatus" : "N/A",
         "RevisionNumber" : 1,
         "SoftwareClassifications" : [ "Firmware/BIOS" ],
         "TargetedOperatingSystems" : "N/A",
         "VersionString" : "M11KT45A"
      },
      {
         "@odata.id" : "/DASH/v1/SoftwareCollection/Software",
         "@odata.type" : "#Software.v1_0_0.Software",
         "BuildNumber" : "N/A",
         "ElementName" : "Dash Firmware",
         "IdentityInfoType" : [ "Dash Firmware" ],
         "IdentityInfoValue" : [ "DASH:Firmware:1" ],
         "InstanceID" : "SW:1",
         "InstanceNumber" : 1,
         "IsEntity" : "True",
         "MajorVersion" : 2,
         "Manufacturer" : "Realtek",
         "MinorVersion" : 110,
         "Name" : "Software",
         "OperationalStatus" : "N/A",
         "RevisionNumber" : 0,
         "SoftwareClassifications" : [ "Firmware" ],
         "TargetedOperatingSystems" : "N/A",
         "VersionString" : "20180822.1200000110"
      }
   ],
   "Name" : "Software Collection"
}
```


Output Class

```
public class SoftwareCollection {
    [JsonProperty("@odata.id")]
    public string _odata_id { get; set; }
    [JsonProperty("@odata.type")]
    public string _odata_type { get; set; }
    public string Id { get; set; }
    [JsonProperty("Members")]
    public List<SoftwareMember> Members { get; set; }
    public string Name { get; set; }
}
public class SoftwareMember {
    [JsonProperty("@odata.id")]
    public string _odata_id { get; set; }
    [JsonProperty("@odata.type")]
    public string odatatype { get; set; }
    public int BuildNumber { get; set; }
    public string ElementName { get; set; }
    public List<string> IdentityInfoType { get; set; }
    public List<string> IdentityInfoValue { get; set; }
    public string InstanceID { get; set; }
```

	public int InstanceNumber { get; set; } public string IsEntity { get; set; } public int MajorVersion { get; set; } public string Manufacturer { get; set; } public int MinorVersion { get; set; } public string Name { get; set; } public List<string> OperationalStatus { get; set; } public int RevisionNumber { get; set; } public List<string> SoftwareClassifications { get; set; } public string TargetedOperatingSystems { get; set; } public string VersionString { get; set; } }

12.10 Enumerate User

This API retrieves user account information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] enumerate user`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["enumerate","user"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/AccountService/ManagerAccountCollection",
	"@odata.type" :
	"#ManagerAccountCollection.v1_0_0.ManagerAccountCollection",
	"Id" : "ManagerAccountCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/ManagerAccountCollection/ManagerAccount",
	"@odata.type" : "#ManagerAccount.v1_0_0.ManagerAccount",
	"AssociatedRoles" : ["Administrator Role"],
	"ElementName" : "Administrator Account",
	"EnabledState" : "Enabled",
	"InstanceNumber" : 0,
	"Name" : "User:1",
	"OrganizationName" : ["My Company"],
	"RequestedState" : "Enabled",
	"UserId" : "Administrator"
	},
	{
	"@odata.id" : "/DASH/v1/ManagerAccountCollection/ManagerAccount",
	"@odata.type" : "#ManagerAccount.v1_0_0.ManagerAccount",
	"AssociatedRoles" : ["Operator Role"],
	"ElementName" : "Operator Account",
	"EnabledState" : "Offline",
	"InstanceNumber" : 1,
	"Name" : "User:2",
	"OrganizationName" : ["My Company"],
	"RequestedState" : "Offline",
	"UserId" : "Operator"
	}
	],
	"Name" : "Manager Account Collection"
	}

Output Class	public class UserCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	[JsonProperty("Members")]
	public List<UserMember> Members { get; set; }
	public string Id { get; set; }
	public string Name { get; set; }
	}
	public class UserMember {

	public List<string> AssociatedRoles { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string Name { get; set; } public List<string> OrganizationName { get; set; } public string RequestedState { get; set; } public string UserId { get; set; } }

12.11 Power Status

This API retrieves the current power state of the target system using the Computer System resource.

DASH CLI Command	**dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power status**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power","status"]}**
Output JSON	**{ "@odata.id" : "/DASH/v1/Systems/12345678-1234-5678-90ab-cddeefaabbcc", "@odata.type" : "#ComputerSystem.v1_5_1.ComputerSystem", "Id" : "12345678-1234-5678-90ab-cddeefaabbcc", "Name" : "Computer System", "PowerState" : "On" }**

Output Class	public class PowerStatus {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	public string Name { get; set; }
	public string PowerState { get; set; }
	}

12.12 Power Available States

This API retrieves the power states that can be requested for the target system as list of strings with power command and description.

DASH CLI Command	**dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power availablestates**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power","availablestates"]}**
Output JSON	**{ "Available Power States (Code - Name)" : ["5 - Power Cycle (Off - Soft)", "8 - Off - Soft", "10 - Master Bus Reset"] }**

Output Class	public class AvailablePowerStates {
	[JsonProperty("Available Power States (Code - Name)")]
	public List<string> AvailablePowerStatesCodeName { get; set; }
	}

12.13 Power Available States 2

This API retrieves the power states that can be requested for the target system with Power state code, Power state description and Power State Name enumerated separately.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power availablestates2`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power","availablestates2"]}`
Output JSON	{ "Available Power States" : [{ "Code" : 3, "Description" : "Initiate the transition of the system to standby or sleep state (corresponding ACPI state G1/S1 or G1/S2).", "Name" : "Sleep - Light" }, { "Code" : 4, "Description" : "Initiate the transition of the system to standby or sleep state (corresponding ACPI state G1/S3).", "Name" : "Sleep -Deep" }, { "Code" : 5, "Description" : "Transition the system to off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power, followed by a transition to on state (corresponding ACPI state G0/S0).", "Name" : "Power Cycle (Off - Soft)" }, { "Code" : 6, "Description" : "Initiate the transition of the system to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the real-time clock.", "Name" : "Off - Hard" }, { "Code" : 7, "Description" : "Transition the system to hibernation state (corresponding ACPI state G1/S4) - write system context to non-volatile storage, power off the system and devices.", "Name" : "Hibernate (Off - Soft)" }, { "Code" : 10, "Description" : "Perform hardware reset on the system.", "Name" : "Master Bus Reset" }, { "Code" : 12, "Description" : "Perform an orderly transition to power off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power.", "Name" : "Off - Soft Graceful" }, { "Code" : 13, "Description" : "Perform an orderly transition to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the real-time clock.", "Name" : "Off - Hard Graceful" }, { "Code" : 15, "Description" : "Perform an orderly transition of the system to power off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power, followed by a transition to on state (corresponding ACPI state G0/S0).",

	"Name" : "Power Cycle (Off - Soft Graceful)"
	},
	{
	"Code" : 16,
	"Description" : "Perform an orderly transition of the system to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the realtime clock, followed by a transition to on state (corresponding ACPI state G0/S0).",
	"Name" : "Power Cycle (Off - Hard Graceful)"
	}
	]
	}

Output Class	public class AvailablePowerStates2 {
	[JsonProperty("Available Power States")]
	public List<AvailablePowerState2> AvailablePowerStatesCodeName2 { get; set; }
	}
	
	public class AvailablePowerState2 {
	public int Code { get; set; }
	public string Name { get; set; }
	}

12.14 Power Supported States

This API retrieves the power states supported by the target system hardware as list of strings with power command and description.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power supportedstates`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power","supportedstates"]}

Output JSON	{
	"Supported Power States (Code - Name)" : [
	"1 - Other",
	"2 - On",
	"3 - Sleep - Light",
	"4 - Sleep -Deep",
	"5 - Power Cycle (Off - Soft)",
	"7 - Hibernate (Off - Soft)",
	"8 - Off - Soft",
	"10 - Master Bus Reset",
	"12 - Off - Soft Graceful",
	"14 - Master Bus Reset Graceful",
	"15 - Power Cycle (Off - Soft Graceful)"
	]
	}

Output Class	public class SupportedPowerStates {
	[JsonProperty("Supported Power States (Code - Name)")]
	public List<string> SupportedPowerStatesCodeName { get; set; }
	}

12.15 Power Supported States 2

This API retrieves the power states supported by the target system hardware with Power state code, Power state description and Power State Name enumerated separately.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power supportedstates2`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power","supportedstates2"]}
Output JSON	{ "Supported Power States" : [{

```
      "Code" : 2,
      "Description" : "Initiate the transition of the system to full on state
(corresponding ACPI state G0/S0).",
      "Name" : "On"
    },
    {
      "Code" : 3,
      "Description" : "Initiate the transition of the system to standby or
sleep state (corresponding ACPI state G1/S1 or G1/S2).",
      "Name" : "Sleep - Light"
    },
    {
      "Code" : 4,
      "Description" : "Initiate the transition of the system to standby or
sleep state (corresponding ACPI state G1/S3).",
      "Name" : "Sleep -Deep"
    },
    {
      "Code" : 5,
      "Description" : "Transition the system to off state (corresponding ACPI
state G2/S5), in which the system consumes a minimal amount of power, followed
by a transition to on state (corresponding ACPI state G0/S0).",
      "Name" : "Power Cycle (Off - Soft)"
    },
    {
      "Code" : 6,
      "Description" : "Initiate the transition of the system to power off state
(corresponding ACPI state G3), in which the power consumption is zero except
for the real-time clock.",
      "Name" : "Off - Hard"
    },
    {
      "Code" : 7,
      "Description" : "Transition the system to hibernation state
(corresponding ACPI state G1/S4) - write system context to non-volatile
storage, power off the system and devices.",
      "Name" : "Hibernate (Off - Soft)"
    },
    {
      "Code" : 10,
      "Description" : "Perform hardware reset on the system.",
      "Name" : "Master Bus Reset"
    },
    {
      "Code" : 12,
      "Description" : "Perform an orderly transition to power off state
(corresponding ACPI state G2/S5), in which the system consumes a minimal amount
of power.",
      "Name" : "Off - Soft Graceful"
    },
    {
      "Code" : 13,
      "Description" : "Perform an orderly transition to power off state
(corresponding ACPI state G3), in which the power consumption is zero except
for the real-time clock.",
      "Name" : "Off - Hard Graceful"
    },
    {
      "Code" : 15,
      "Description" : "Perform an orderly transition of the system to power off
state (corresponding ACPI state G2/S5), in which the system consumes a minimal
amount of power, followed by a transition to on state (corresponding ACPI state
G0/S0).",
      "Name" : "Power Cycle (Off - Soft Graceful)"
    },
    {
      "Code" : 16,
      "Description" : "Perform an orderly transition of the system to power off
state (corresponding ACPI state G3), in which the power consumption is zero
```

	except for the realtime clock, followed by a transition to on state (corresponding ACPI state G0/S0).", "Name" : "Power Cycle (Off - Hard Graceful)" }] }

Output Class	public class SupportedPowerStates2 {
	[JsonProperty("Supported Power States")]
	public List<SupportedPowerState2> SupportedPowerStatesCodeName2 { get; set; }
	}
	public class SupportedPowerState2 {
	public int Code { get; set; }
	public string Name { get; set; }
	}

12.16 Power State Change

This API initiates a power state transition for the target system.

	<<option>>	
DASH CLI Command		dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power <<option>>
jdo Usage		dashcli -jdo
URI		https://<AMA_HOST>:<AMA_PORT>/
Input JSON		{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power","<<option>>"]}

Output JSON	on	{
		"ErrorCode" : 0,
		"ErrorMessage" : "On Reset Succeeded"
		}

	off	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Off Reset Succeeded"
		}

	cycle	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Cycle Reset Succeeded"
		}

	restart	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Graceful Reset Succeeded"
		}

	reset	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Hard Reset Succeeded"
		}

	offgraceful	{
		"ErrorCode" : -1,
		"ErrorMessage" : "Error: Power Operation Failed"
		}

	sleep	{
		"ErrorCode" : -1,
		"ErrorMessage" : "Error: Power Operation Failed"
		}

	hibernate	{
		"ErrorCode" : -1,
		"ErrorMessage" : "Error: Power Operation Failed"
		}

Output Class		public class CommandError {
		public int ErrorCode { get; set; }
		public string ErrorMessage { get; set; }
		}

Note: List of Power States (Refer DSP 1027)

<<option>> Name	<<option>> Code	Description

on	2	Initiate the transition of the system to full on state.
	3	Initiate the transition of the system to standby or sleep state.
sleep	4	Initiate the transition of the system to standby or sleep state.
cycle/softreset	5	Transition the system to off state in which the system consumes a minimal amount of power, followed by a transition to on state.
	6	Initiate the transition of the system to power off state, in which the power consumption is zero except for the real-time clock.
hibernate	7	Transition the system to hibernation state. – write system context to non-volatile storage, power off the system and devices.
turnoff/off	8	Initiate the transition of the system to off state, in which the system consumes a minimal amount of power.
	9	Transition the system to power off state, in which the power consumption is zero except for the real-time clock, followed by a transition to on state.
reset/hardreset	10	Perform hardware reset on the system.
	11	Assert an NMI on the system.
shutdown/offgraceful	12	Perform an orderly transition to power off state, in which the system consumes a minimal amount of power.
	13	Perform an orderly transition to power off state, in which the power consumption is zero except for the real-time clock.
restart/resetgraceful	14	Perform an orderly shutdown of the system followed by hardware reset.
	15	Perform an orderly transition of the system to power off state, in which the system consumes a minimal amount of power, followed by a transition to on state.
	16	Perform an orderly transition of the system to power off state, in which the power consumption is zero except for the real-time clock, followed by a transition to on state.

12.17 Power State String

This API retrieves the string representation of the current power state.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power powerstatestring`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power"," powerstatestring"]}`

Output JSON

```
{
  "Power States" :
  [
    {
      "Code" : 1,
      "Name" : "Other"
    },
    {
      "Code" : 2,
      "Name" : "On"
    },
    {
      "Code" : 3,
      "Name" : "Sleep - Light"
    },
    {
      "Code" : 4,
      "Name" : "Sleep -Deep"
    },
    {
      "Code" : 5,
      "Name" : "Power Cycle (Off - Soft)"
    },
    {
      "Code" : 6,
      "Name" : "Off - Hard"
    },
    {
      "Code" : 7,
      "Name" : "Hibernate (Off - Soft)"
    },
    {
      "Code" : 8,
      "Name" : "Off - Soft"
    },
    {
      "Code" : 9,
      "Name" : "Power Cycle (Off-Hard)"
    },
    {
      "Code" : 10,
      "Name" : "Master Bus Reset"
    },
    {
      "Code" : 11,
      "Name" : "Diagnostic Interrupt (NMI)"
    },
    {
      "Code" : 12,
      "Name" : "Off - Soft Graceful"
    },
    {
      "Code" : 13,
      "Name" : "Off - Hard Graceful"
    },
    {
      "Code" : 14,
      "Name" : "Master Bus Reset Graceful"
    },
    {
      "Code" : 15,
      "Name" : "Power Cycle (Off - Soft Graceful)"
    },
    {
      "Code" : 16,
      "Name" : "Power Cycle (Off - Hard Graceful)"
    }
  ]
}
```

Output Class	public class PowerStateString
	{
	public PowerStateStringMember [] PowerStates { get; set; }
	}
	
	public class PowerStateStringMember
	{
	public int Code { get; set; }
	public string Name { get; set; }
	}

12.18 Power State Description

This API retrieves a human-readable description of the current power state.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t computersystem[0] power powerstatedescription
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["power"," powerstatedescription"]}
Output JSON	{ "Power State Descriptions" : [{ "Code" : 1, "Name" : "-" }, { "Code" : 2, "Name" : "Initiate the transition of the system to full on state (corresponding ACPI state G0/S0)." }, { "Code" : 3, "Name" : "Initiate the transition of the system to standby or sleep state (corresponding ACPI state G1/S1 or G1/S2)." }, { "Code" : 4, "Name" : "Initiate the transition of the system to standby or sleep state (corresponding ACPI state G1/S3)." }, { "Code" : 5, "Name" : "Transition the system to off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power, followed by a transition to on state (corresponding ACPI state G0/S0)." }, { "Code" : 6, "Name" : "Initiate the transition of the system to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the real-time clock." }, { "Code" : 7, "Name" : "Transition the system to hibernation state (corresponding ACPI state G1/S4) - write system context to non-volatile storage, power off the system and devices." }, { "Code" : 8, "Name" : "Initiate the transition of the system to off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power." },

	{
	"Code" : 9,
	"Name" : "Transition the system to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the real-time clock, followed by a transition to on state (corresponding ACPI state G0/S0)."
	},
	{
	"Code" : 10,
	"Name" : "Perform hardware reset on the system."
	},
	{
	"Code" : 11,
	"Name" : "Assert an NMI on the system."
	},
	{
	"Code" : 12,
	"Name" : "Perform an orderly transition to power off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power."
	},
	{
	"Code" : 13,
	"Name" : "Perform an orderly transition to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the real-time clock."
	},
	{
	"Code" : 14,
	"Name" : "Perform an orderly shutdown of the system followed by hardware reset."
	},
	{
	"Code" : 15,
	"Name" : "Perform an orderly transition of the system to power off state (corresponding ACPI state G2/S5), in which the system consumes a minimal amount of power, followed by a transition to on state (corresponding ACPI state G0/S0)."
	},
	{
	"Code" : 16,
	"Name" : "Perform an orderly transition of the system to power off state (corresponding ACPI state G3), in which the power consumption is zero except for the realtime clock, followed by a transition to on state (corresponding ACPI state G0/S0)."
	}
	]
	}

Output Class	`public class PowerStateDescription`
	`{`
	`public PowerStateDescriptionMember[] PowerStateDescriptions{ get; set; }`
	`}`
	
	`public class PowerStateDescriptionMember`
	`{`
	`public int Code { get; set; }`
	`public string Name { get; set; }`
	`}`

12.19 Asset

This API retrieves asset information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ asset[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/asset[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
	"@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
	"CanbeFRUed" : "No",
	"ElementName" : "Processor #1",
	"InstanceNumber" : 0,
	"Manufacturer" : "AMD",
	"Model" : "AMD Phenom(tm) II X4 B95 Processor",
	"Name" : "PhysicalAsset",
	"PartNumber" : "N/A",
	"SKU" : "N/A",
	"SerialNumber" : "N/A",
	"Tag" : "BRCM:4.1"
	}

Output Class	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }
	public int InstanceNumber { get; set; }
	public string Manufacturer { get; set; }
	public string Model { get; set; }
	public string Name { get; set; }
	public string PartNumber { get; set; }
	public string SKU { get; set; }
	public string SerialNumber { get; set; }
	public int SlotNumber { get; set; }
	public string Tag { get; set; }
	public string CanbeFRUed { get; set; }
	public string MemoryBankLabel { get; set; }
	public string MemoryCapacity { get; set; }
	public string MemoryFormFactor { get; set; }
	public string MemoryType { get; set; }
	public string IsHostingBoard { get; set; }
	public string PackageType { get; set; }
	public string ChassisType { get; set; }
	}

12.20 Boot Config

This API retrieves boot configuration information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ bootconfig[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/bootconfig[0]", "Commands":["show"]}
Output JSON	{ "@odata.id" : "/DASH/v1/BootConfigurationCollection/BootConfiguration", "@odata.type" : "#BootConfiguration.v1_0_0.BootConfiguration", "BootDevices" : [{ "Name" : "BRCM:Network:3", "Value" : 0 }, { "Name" : "BRCM:Hard Drive:1", "Value" : 1 }, { "Name" : "BRCM:USB Device:4", "Value" : 2

	`}, { "Name" : "BRCM:CD-ROM:2", "Value" : 3 }], "ElementName" : "Boot Configuration Setting #1", "Id" : "BootConfiguration", "InstanceId" : "BRCM:31.1", "InstanceNumber" : 0, "IsCurrentConfiguration" : "No", "IsDefaultConfiguration" : "No", "IsNextConfiguration" : "Yes", "Name" : "Boot Configuration" }`
Output Class	`public class BootConfigurationMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public List<BootDevice> BootDevices { get; set; } public string ElementName { get; set; } public string Id { get; set; } public string InstanceId { get; set; } public int InstanceNumber { get; set; } public string IsCurrentConfiguration { get; set; } public string IsDefaultConfiguration { get; set; } public string IsNextConfiguration { get; set; } public string Name { get; set; } } public class BootDevice { public string Name { get; set; } public int Value { get; set; } }`

Note: List of Boot Config commands

Command	Input JSON command
Show	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/bootconfig[0]", "Commands":["show"]}`
Change Boot Order	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/bootconfig[0]", "Commands":["changebootorder","3","2","1","0"]}`

12.21 Fan

This API retrieves fan information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ fan[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/fan[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/FanCollection/Fan", "@odata.type" : "#Fan.v1_0_0.Fan", "DesiredSpeed" : "N/A", "DeviceID" : "BRCM:10.1", "ElementName" : "Fan #1", "EnabledState" : "Not Applicable", "HealthState" : "OK", "InstanceNumber" : 0, "IsActiveCooling" : "Yes", "IsVariableSpeed" : "Yes", "Name" : "Fan", "OperationalState" : ["OK"], "RequestedState" : "Not Applicable" }`

Output Class	public class FanMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public int DesiredSpeed { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string IsActiveCooling { get; set; }
	public string IsVariableSpeed { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public string RequestedState { get; set; }
	}

Note: List of Fan commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/fan[0]", "Commands":["show"]}
Enumerate Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/fan[0]", "Commands":["enumerate","asset"]}
Asset Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/fan[0]/asset[0]", "Commands":["show"]}

12.22 Memory

This API retrieves memory information associated with the Computer System resource.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ memory[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/memory[0]", "Commands":["show"]}

Output JSON	{
	"@odata.id" : "/DASH/v1/MemoryCollection/Memory",
	"@odata.type" : "#Memory.v1_0_0.Memory",
	"AccessType" : "Read/Write Supported",
	"AvailableMemory" : "4096 MB",
	"DeviceID" : "BRCM:8.1",
	"ElementName" : "Total System Memory",
	"EnabledState" : "Enabled",
	"HealthState" : "Unknown",
	"InstanceNumber" : 0,
	"IsVolatileMemory" : "Yes",
	"Name" : "Memory",
	"RequestedState" : "Not Applicable",
	"TotalMemory" : "4096 MB"
	}

Output Class	public class MemoryMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string AccessType { get; set; } public string AvailableMemory { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string IsVolatileMemory { get; set; } public string Name { get; set; }

	public string RequestedState { get; set; } public string TotalMemory { get; set; } }

Note: List of Memory commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/memory[0]", "Commands":["show"]}
Enumerate Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/memory[0]", "Commands":["enumerate","asset"]}
Asset Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/memory[0]/asset[0]", "Commands":["show"]}

12.23 Power Supply

This API retrieves power supply information associated with the Computer System resource.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ powersupply[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/powersupply[0]", "Commands":["show"]}

Output JSON	{
	"@odata.id" : "/DASH/v1/PowerSupplyCollection/PowerSupply",
	"@odata.type" : "#PowerSupply.v1_0_0.PowerSupply",
	"DeviceID" : "BRCM:9.1",
	"ElementName" : "Power Supply #1",
	"EnabledState" : "Enabled",
	"HealthState" : "Unknown",
	"InstanceNumber" : 0,
	"Name" : "Power Supply",
	"OperationalState" : ["Unknown"],
	"RequestedState" : "Not Applicable",
	"TotalPower" : 0
	}

Output Class	public class PowerSupplyMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public string RequestedState { get; set; }
	public int TotalPower { get; set; }
	}

Note: List of Power Supply commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/powersupply[0]", "Commands":["show"]}
Enumerate Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/powersupply[0]", "Commands":["enumerate","asset"]}
Asset Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/powersupply[0]/asset[0]", "Commands":["show"]}

12.24 Processors

This API retrieves processor information associated with the Computer System resource.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/processor[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]","Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/ProcessorCollection/Processor",
	"@odata.type" : "#Processor.v1_0_0.Processor",
	"CPUStatus" : "CPU Enabled",
	"CurClockSpeed" : "3000 MHz",
	"DeviceID" : "BRCM:7.1",
	"ElementName" : "Processor #1",
	"EnabledState" : "Enabled",
	"ExtBusSpeed" : "200 MHz",
	"Family" : "AMD Phenom(TM) II Processor Family",
	"HealthState" : "Unknown",
	"InstanceNumber" : 0,
	"LoadPercent" : "N/A",
	"MaxClockSpeed" : "3000 MHz",
	"Name" : "Processor",
	"OperationalState" : ["Unknown"],
	"RequestedState" : "Not Applicable"
	}

Output Class	public class ProcessorMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string CPUStatus { get; set; }
	public string CurClockSpeed { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string ExtBusSpeed { get; set; }
	public string Family { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string LoadPercent { get; set; }
	public string MaxClockSpeed { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public string RequestedState { get; set; }
	}

Note: List of Processor commands

Command	Input JSON command
Show	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]","Commands":["show"]}`
Enumerate Asset	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]","Commands":["enumerate","asset"]}`
Enumerate Fan	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]","Commands":["enumerate","fan"]}`
Enumerate Sensor	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]","Commands":["enumerate","sensor"]}`
Enumerate Memory	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]","Commands":["enumerate","memory"]}`
Asset Show	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/asset[0]","Commands":["show"]}`

Fan Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/fan[0]", "Commands":["show"]}
Fan Enumerate Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/fan[0]", "Commands":["enumerate","asset"]}
Fan Show Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/fan[0]/asset[0]", "Commands":["show"]}
Memory Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/memory[0]", "Commands":["show"]}
Memory Enumerate Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/memory[0]", "Commands":["enumerate","asset"]}
Memory Show Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/memory[0]/asset[0]", "Commands":["show"]}
Sensor Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/sensor[0]", "Commands":["show"]}
Sensor Set Threshold	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/processor[0]/sensor[0]", "Commands":["setthreshold","lnc","100"]}

12.25 Sensor

This API retrieves sensor information associated with the Computer System resource.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ sensor[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/sensor[0]", "Commands":["show"]}
Output JSON	{ "@odata.id" : "/DASH/v1/SensorCollection/Sensor", "@odata.type" : "#Sensor.v1_0_0.Sensor", "BaseUnit" : "RPM", "CurrentState" : "Normal", "DeviceID" : "BRCM:11.1", "ElementName" : "Numeric Sensor #1 (Tachometer)", "EnabledState" : "Not Applicable", "HealthState" : "Unknown", "InstanceNumber" : 0, "Name" : "Sensor", "OperationalState" : ["No Contact"], "PossibleStates" : ["Non-Critical", "Lower Non-Critical", "Upper Non-Critical", "Critical", "Lower Critical", "Upper Critical", "Fatal", "Lower Fatal", "Upper Fatal", "Normal", "Unknown"], "RateUnits" : "None", "RequestedState" : "No Change", "SensorReading" : 0, "SensorType" : "Tachometer", "ThresholdLC" : "N/A", "ThresholdLF" : "N/A", "ThresholdLNC" : "N/A", "ThresholdUC" : "N/A", "ThresholdUF" : "N/A",

	"ThresholdUNC" : "N/A", "UnitModifier" : 0 }

Output Class	public class SensorMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string BaseUnit { get; set; }
	public string CurrentState { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public List<string> PossibleStates { get; set; }
	public string RateUnits { get; set; }
	public string RequestedState { get; set; }
	public int SensorReading { get; set; }
	public string SensorType { get; set; }
	public int ThresholdLC { get; set; }
	public int ThresholdLF { get; set; }
	public int ThresholdLNC { get; set; }
	public int ThresholdUC { get; set; }
	public int ThresholdUF { get; set; }
	public int ThresholdUNC { get; set; }
	public int UnitModifier { get; set; }
	}

Note: List of Sensor commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/sensor[0]", "Commands":["show"]}
Reading	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/sensor[0]", "Commands":["reading"]}
Set Threshold	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/sensor[0]", "Commands":["setthreshold","lnc","100"]}

12.26 Software

This API retrieves software information associated with the Computer System resource.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/ software[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/software[0]", "Commands":["show"]}
Output JSON	{ "@odata.id" : "/DASH/v1/SoftwareCollection/Software", "@odata.type" : "#Software.v1_0_0.Software", "BuildNumber" : 1, "ElementName" : "Bios Firmware", "IdentityInfoType" : ["Bios Firmware"], "IdentityInfoValue" : ["BIOS:285409537:0"], "InstanceID" : "SW:0", "InstanceNumber" : 0, "IsEntity" : "True", "MajorVersion" : 1, "Manufacturer" : "LENOVO", "MinorVersion" : 0, "Name" : "Software", "OperationalStatus" : "N/A", "RevisionNumber" : 1, "SoftwareClassifications" : ["Firmware/BIOS"],

	"TargetedOperatingSystems" : "N/A", "VersionString" : "M11KT45A" }

Output Class	public class SensorMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string BaseUnit { get; set; }
	public string CurrentState { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public List<string> PossibleStates { get; set; }
	public string RateUnits { get; set; }
	public string RequestedState { get; set; }
	public int SensorReading { get; set; }
	public string SensorType { get; set; }
	public int ThresholdLC { get; set; }
	public int ThresholdLF { get; set; }
	public int ThresholdLNC { get; set; }
	public int ThresholdUC { get; set; }
	public int ThresholdUF { get; set; }
	public int ThresholdUNC { get; set; }
	public int UnitModifier { get; set; }
	}

Note: List of Software commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/software[0]", "Commands":["show"]}
Install	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/software[0]", "Commands":["install","http://10.138.141.209/xampp/bin/5762tm3.05"]}

12.27 User Management

These APIs can be used to perform user management operations such as addition/enable/ disable/ delete operations on user as well as association/disassociation of roles with users.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t computersystem[0]/user[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[0]", "Commands":["show"]}

Output JSON	{
	"@odata.id" : "/DASH/v1/ManagerAccountCollection/ManagerAccount",
	"@odata.type" : "#ManagerAccount.v1_0_0.ManagerAccount",
	"AssociatedRoles" : ["Administrator Role"],
	"ElementName" : "Administrator Account",
	"EnabledState" : "Enabled",
	"InstanceNumber" : 0,
	"Name" : "User:1",
	"OrganizationName" : ["My Company"],
	"RequestedState" : "Enabled",
	"UserId" : "Administrator"
	}

Output Class	public class UserMember { public List<string> AssociatedRoles { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string Name { get; set; } public List<string> OrganizationName { get; set; }

```
        public string RequestedState { get; set; }
        public string UserId { get; set; }
    }
```

Note: List of User commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[2]", "Commands":["show"]}
Enable	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[2]", "Commands":["enable"]}
Disable	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[2]", "Commands":["disable"]}
Assign Roles	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[2]", "Commands":["assignroles","Role:0","Role:1"]}
Remove Roles	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[2]", "Commands":["removeroles","Role:1"]}
Delete	{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]/user[0]", "Commands":["delete"]}

Note: Refer section User for User profile

13 DHCP Client

Purpose: DHCP client configuration management.

13.1 DHCP Client Show

Shows detailed DHCP client configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t dhcpclient[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"dhcpclient[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/DHCPClientCollection/DHCPClient", "@odata.type" : "#DHCPClient.v1_0_0.DHCPClient", "ClientState" : "Unknown", "ElementName" : "DHCP:0", "EnabledState" : "Enabled", "InstanceNumber" : 0, "Name" : "DHCP Client", "NameFormat" : "2", "OtherTypeDescription" : "DHCP", "ProtocolIFType" : "Other", "RequestedState" : "Enabled" }`
Output Class	`public class DHCPClientMember { public string odataid { get; set; } public string odatatype { get; set; } public string ClientState { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string NameFormat { get; set; } public string OtherTypeDescription { get; set; } public string ProtocolIFType { get; set; } public string RequestedState { get; set; } }`

14 Disk Information

Purpose: Physical Disk configuration management.

14.1 Disk Information Show

Shows detailed Disk configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t diskinfo[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"diskinfo[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/DiskInfoCollection", "@odata.type" : "#DiskInfoCollection.v1_5_1.DiskInfoCollection", "Id" : "DiskInfoCollection", "Members" : [{ "@odata.id" : "/DASH/v1/DiskInfoCollection/DiskInfo", "@odata.type" : "#DiskInfo.v1_0_0.DiskInfo", "DeviceID" : "DiskDrive:0", "DiskCapacity" : "512 GB", "DiskModelNumber" : "SAMSUNG MZVL2512HCJQ-00B00 ", "DiskSerialNumber" : "S675NE0R100598 ", "DiskTypeName" : "NVME", "ElementName" : "DiskDrive", "InstanceID" : "AMD:36:0", "InstanceNumber" : 0, "Name" : "DiskDrive", "SystemName" : "DASHStack" }], "Name" : "DiskInfo Collection" }`
Output Class	`public class DiskInfo { public string odataid { get; set; } public string odatatype { get; set; } public string DeviceID { get; set; } public string DiskCapacity { get; set; } public string DiskModelNumber { get; set; } public string DiskSerialNumber { get; set; } public string DiskTypeName { get; set; } public string ElementName { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string SystemName { get; set; } }`

Disk Information wipedisk Actions

Performs secure disk wipe operations on a specified disk.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t diskinfo[0] wipedisk
jdo Usage	dashcli -jdo {"h":"dash-system","u":"admin","P":"adminpass","t":"diskinfo[0]", "Commands":["wipedisk"]}
URI	https://<AMA_HOST>:<AMA_PORT>/

Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"diskinfo[0]", "Commands":["wipedisk"]}
Output JSON	{ "ErrorCode" : 0, "ErrorMessage" : "Disk wipe initiated successfully" }
Output Class	public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }

15 DNS Client

Purpose: DNS client configuration management.

15.1 DNS Client Show

Shows detailed DNS client configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t dnsclient[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"dnsclient[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/DNSClientCollection/DNSClient",
	"@odata.type" : "#DNSClient.v1_0_0.DNSClient",
	"DHCPOptionsToUse" : "N/A",
	"DNSSuffixesToAppend" : "N/A",
	"DomainName" : "amd.com",
	"ElementName" : "DNS:0",
	"EnabledState" : "Enabled",
	"Hostname" : "BLR-5CG012495X",
	"InstanceNumber" : 0,
	"Name" : "DNS Client",
	"NameFormat" : "2",
	"ProtocolIFType" : "Other",
	"RequestedState" : "No Change",
	"appendParentSuffixes" : "N/A",
	"appendPrimarySuffixes" : "N/A",
	"registerThisConnectionsAddress" : "N/A",
	"useSuffixWhenRegistering" : "N/A"
	}

Output Class	public class DNSClientMember
	{
	public string odataid { get; set; }
	public string odatatype { get; set; }
	public string DHCPOptionsToUse { get; set; }
	public string DNSSuffixesToAppend { get; set; }
	public string DomainName { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string Hostname { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public string NameFormat { get; set; }
	public string ProtocolIFType { get; set; }
	public string RequestedState { get; set; }
	public string appendParentSuffixes { get; set; }
	public string appendPrimarySuffixes { get; set; }
	public string registerThisConnectionsAddress { get; set; }
	public string useSuffixWhenRegistering { get; set; }
	}

16 Ethernet Port

Purpose: Ethernet network port configuration and status.

16.1 Ethernet Port Show

Shows detailed information for a specific Ethernet port.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t ethernetport[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"ethernetport[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id": "/DASH/v1/EthernetPortCollection/EthernetPort", "@odata.type": "#EthernetPort.v1_0_0.EthernetPort", "Capabilities": ["Unknown"], "DeviceID": "26", "ElementName": "EthernetPort:0", "EnabledCapabilities": ["Unknown"], "EnabledState": "Enabled", "InstanceNumber": 0, "LinkTechnology": "Ethernet", "MACAddress": "00:00:1a:20:6a:bf", "MaximumSpeed": "N/A", "Name": "Realtek PCIe GbE Family Controller", "NetworkAddress": ["00000A560FC9"], "PortType": "Unknown", "RequestedSpeed": "N/A", "RequestedState": "Not Applicable", "Speed": "1 Gbps" }`
Output Class	`public class EthernetPortMember { public string odataid { get; set; } public string odatatype { get; set; } public List <string> Capabilities { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public List <string> EnabledCapabilities { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string LinkTechnology { get; set; } public string MACAddress { get; set; } public string MaximumSpeed { get; set; } public string Name { get; set; } public List <string> NetworkAddress { get; set; } public string PortType { get; set; } public string RequestedSpeed { get; set; } public string RequestedState { get; set; } public string Speed { get; set; } }`

17 Fan

Purpose: Fan device status and management

17.1 Fan Show

Shows detailed information for a specific fan instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t fan[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"fan[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/FanCollection/Fan", "@odata.type" : "#Fan.v1_0_0.Fan", "DesiredSpeed" : "N/A", "DeviceID" : "BRCM:10.1", "ElementName" : "Fan #1", "EnabledState" : "Not Applicable", "HealthState" : "OK", "InstanceNumber" : 0, "IsActiveCooling" : "Yes", "IsVariableSpeed" : "Yes", "Name" : "Fan", "OperationalState" : ["OK"], "RequestedState" : "Not Applicable" }`
Output Class	`public class FanMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public int DesiredSpeed { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string IsActiveCooling { get; set; } public string IsVariableSpeed { get; set; } public string Name { get; set; } public List<string> OperationalState { get; set; } public string RequestedState { get; set; } }`

17.2 Enumerate Asset

Lists assets associated with a specific fan.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t fan[0] enumerate asset`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"fan[0]", "Commands":["enumerate","asset"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection", "@odata.type" : "#PhysicalAssetCollection.v1_0_0.PhysicalAssetCollection", "Id" : "PhysicalAssetCollection", "Members" : [{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset", "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset", "CanbeFRUed" : "Yes", "ElementName" : "BaseBoard:0", "InstanceNumber" : 0, "IsHostingBoard" : "Yes", "Manufacturer" : "LENOVO",`

```
            "Model" : "30FD",
            "Name" : "PhysicalAsset",
            "PartNumber" : "SDK0J40697 WIN 3305035440590",
            "SKU" : "NULL",
            "SerialNumber" : "                      ",
            "Tag" : "Card:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ChassisType" : "Desktop",
            "ElementName" : "Chassis:0",
            "InstanceNumber" : 1,
            "Manufacturer" : "LENOVO",
            "Model" : "10M30009US",
            "Name" : "PhysicalAsset",
            "PackageType" : "Chassis/Frame",
            "PartNumber" : "None",
            "SKU" : "LENOVO_MT_10M3_BU_LENOVO_FM_ThinkCentre M715q",
            "SerialNumber" : "MJ04WH8F",
            "Tag" : "Chassis:0",
            "Version" : "None"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ElementName" : "Processor:0",
            "InstanceNumber" : 2,
            "Manufacturer" : "AuthenticAMD",
            "Model" : "AMD PRO A12-9800E R7, 12 COMPUTE CORES 4C+8G   ",
            "Name" : "PhysicalAsset",
            "PartNumber" : "UnKnown ",
            "SKU" : "NULL",
            "SerialNumber" : "UnKnown ",
            "Tag" : "Processor:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ElementName" : "Memory:0",
            "InstanceNumber" : 3,
            "Manufacturer" : "Hynix/Hyundai",
            "MemoryBankLabel" : "CHANNEL A",
            "MemoryCapacity" : "8192 MB",
            "MemoryFormFactor" : "DIMM",
            "MemorySpeed" : "2400 nanosecond",
            "MemoryType" : "DDR4",
            "Model" : "N/A",
            "Name" : "PhysicalAsset",
            "PartNumber" : "HMA81GS6AFR8N-UH    ",
            "SKU" : "N/A",
            "SerialNumber" : "51B27A3F",
            "Tag" : "Mem:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:0",
            "InstanceNumber" : 4,
            "Manufacturer" : "N/A",
            "Model" : "J6B2",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
```

<table>
<tr><td></td><td>


```
            "SerialNumber" : "N/A",
            "SlotNumber" : 0,
            "Tag" : "Slot:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:1",
            "InstanceNumber" : 5,
            "Manufacturer" : "N/A",
            "Model" : "J6B1",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "SlotNumber" : 1,
            "Tag" : "Slot:1"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:2",
            "InstanceNumber" : 6,
            "Manufacturer" : "N/A",
            "Model" : "J6D1",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "SlotNumber" : 2,
            "Tag" : "Slot:2"
         }
      ],
      "Name" : "Physical Asset Collection"
   }
```


</td></tr>
<tr><td>Output Class</td><td>

```
    public class PhysicalAssetCollection {
        [JsonProperty("@odata.id")]
        public string _odata_id { get; set; }
        [JsonProperty("@odata.type")]
        public string _odata_type { get; set; }
        public string Id { get; set; }
        [JsonProperty("Members")]
        public List<PhysicalAssetMember> Members { get; set; }
        public string Name { get; set; }
    }
    public class PhysicalAssetMember {
        [JsonProperty("@odata.id")]
        public string _odata_id { get; set; }
        [JsonProperty("@odata.type")]
        public string odatatype { get; set; }
        public string ConnectorLayout { get; set; }
        public List<string> ConnectorType { get; set; }
        public string ElementName { get; set; }
        public int InstanceNumber { get; set; }
        public string Manufacturer { get; set; }
        public string Model { get; set; }
        public string Name { get; set; }
        public string PartNumber { get; set; }
        public string SKU { get; set; }
        public string SerialNumber { get; set; }
        public int SlotNumber { get; set; }
        public string Tag { get; set; }
        public string CanbeFRUed { get; set; }
        public string MemoryBankLabel { get; set; }
```

</td></tr>
</table>

	public string MemoryCapacity { get; set; } public string MemoryFormFactor { get; set; } public string MemoryType { get; set; } public string IsHostingBoard { get; set; } public string PackageType { get; set; } public string ChassisType { get; set; } }

17.3 Asset

Shows a specific asset linked to a fan.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t fan[0]/asset[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"fan[0]/asset[0]", "Commands":["show"]}

Output JSON	{
	"@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
	"@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
	"CanbeFRUed" : "Yes",
	"ElementName" : "BaseBoard:0",
	"InstanceNumber" : 0,
	"IsHostingBoard" : "Yes",
	"Manufacturer" : "LENOVO",
	"Model" : "30FD",
	"Name" : "PhysicalAsset",
	"PartNumber" : "SDK0J40697 WIN 3305035440590",
	"SKU" : "NULL",
	"SerialNumber" : " ",
	"Tag" : "Card:0"
	}

Output Class	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }
	public int InstanceNumber { get; set; }
	public string Manufacturer { get; set; }
	public string Model { get; set; }
	public string Name { get; set; }
	public string PartNumber { get; set; }
	public string SKU { get; set; }
	public string SerialNumber { get; set; }
	public int SlotNumber { get; set; }
	public string Tag { get; set; }
	public string CanbeFRUed { get; set; }
	public string MemoryBankLabel { get; set; }
	public string MemoryCapacity { get; set; }
	public string MemoryFormFactor { get; set; }
	public string MemoryType { get; set; }
	public string IsHostingBoard { get; set; }
	public string PackageType { get; set; }
	public string ChassisType { get; set; }
	}

18 Filter Collection

Purpose: Event filter collections for indication subscriptions.

18.1 Filter Collection Show

Shows details for a specific filter collection.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t filtercollection[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"filtercollection[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/FilterCollectionCollection/FilterCollection", "@odata.type" : "#FilterCollection.v1_0_0.FilterCollection", "CollectionName" : "BRCM:PlatformEvents", "InstanceID" : "BRCM:24.1", "InstanceNumber" : 0, "Name" : "FilterCollection" }`
Output Class	`public class FilterCollectionMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string CollectionName { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } }`

18.2 Filter Collection Show All Filters

Lists all filters within a specific filter collection.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t filtercollection[0] showallfilters`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"filtercollection[0]", "Commands":["showallfilters"]}`
Output JSON	`{ "Members" : [] }`
Output Class	`public class ShowAllFilters { public List<Filter> Members { get; set; } } public class Filter { public string Name { get; set; } public string QueryString { get; set; } }`

18.3 Filter Collection Delete

Deletes a specific filter collection.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t filtercollection[0] delete`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"filtercollection[0]", "Commands":["delete"]}`
Output JSON	`{ "ErrorCode" : -1, "ErrorMessage" : "Error: Deleting Indication Filter" }`
Output Class	`public class CommandError { public int ErrorCode { get; set; }`

	`public string ErrorMessage { get; set; }` `}`

19 Indication

Purpose: Event indication and subscription management.

19.1 Indication Subscribe

Subscribes to event notifications using a filter collection and destination URI.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass indication subscribe`
	`Notification URI [Mandatory]: http://10.136.131.230:8080/eventsink`
	`DELIVERY MODES [Mandatory]: push`
	`Expiration Time [Optional]: NULL`
	`FILTER TYPES [Mandatory]: Filter Collection`
	`Choose valid instance number from the above list : 13`
	`EndTo URI [Default - Delivered to NotifyTo URI]: NULL`
	`Encode type [Optional] : NULL`
	`Locale (Ex: ln-en) [Optional] : NULL`
	`Request BookMarks [Y/N] [Optional] : NULL`
	`Heartbeat Interval in seconds [Optional] : NULL`
	`Connection Retry Count [Optional] : NULL`

jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass",` `"Commands":["indication","subscribe","13","http://10.136.131.230:8080/eventsink"]}`

Output JSON	`{`
	`"ErrorCode" : 0,`
	`"ErrorMessage" : "Indication subscription created successfully"`
	`}`

Output Class	`public class CommandError {`
	`public int ErrorCode { get; set; }`
	`public string ErrorMessage { get; set; }`
	`}`

20 Indication Filter

Purpose: Management of indication filters for event subscriptions.

20.1 Indication Filter Show

Shows details for a specific indication filter.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t indicationfilter[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"indicationfilter[0]", "Commands":["show"]}`

Output JSON	<pre>{
	"@odata.id" : "/DASH/v1/IndicationFilterCollection/IndicationFilter",
	"@odata.type" : "#IndicationFilter.v1_0_0.IndicationFilter",
	"CreationClassName" : "CIM_IndicationFilter",
	"ElementName" : "N/A",
	"IndividualSubscriptionSupported" : "True",
	"InstanceNumber" : 0,
	"Name" : "IndicationFilter",
	"Query" : "SELECT * FROM CIM_AlertIndication",
	"QueryLanguage" : "WQL",
	"SourceNamespace" : "WQL",
	"SystemCreationClassName" : "CIM_IndicationFilter",
	"SystemName" : "ManagedSystem"
	}</pre>

Output Class	<pre>public class IndicationFilterMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string CreationClassName { get; set; }
	public string ElementName { get; set; }
	public string IndividualSubscriptionSupported { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public string Query { get; set; }
	public string QueryLanguage { get; set; }
	public string SourceNamespace { get; set; }
	public string SystemCreationClassName { get; set; }
	public string SystemName { get; set; }
	}</pre>

20.2 Indication Filter Delete

Deletes a specific indication filter.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t indicationfilter[0] delete`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"indicationfilter[0]", "Commands":["delete"]}`

Output JSON	<pre>{
	"ErrorCode" : 0,
	"ErrorMessage" : "Indication Filter Deleted Successfully"
	}</pre>

Output Class	<pre>public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}</pre>

21 Indication Subscription

Purpose: Management of active event subscriptions.

21.1 Indication Subscription Show

Shows details for a specific event subscription.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t indicationsubscription[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"indicationsubscription[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/IndicationSubscriptionCollection/IndicationSubscription", "@odata.type" : "#IndicationSubscription.v1_0_0.IndicationSubscription", "Destination" : "http://127.0.0.1:8080/eventsink", "FailureTriggerTimeInterval" : 0, "FilterName" : "DMTF:INFO", "InstanceNumber" : 0, "Name" : "IndicationSubscription", "OnFatalErrorPolicy" : "Remove", "OtherOnFatalErrorPolicy" : "N/A", "OtherSubscriptionState" : "N/A", "RepeatNotificationCount" : "N/A", "RepeatNotificationGap" : "N/A", "RepeatNotificationInterval" : "N/A", "RepeatNotificationPolicy" : "None", "SubscriptionState" : "Enabled" }`
Output Class	`public class IndicationSubscriptionMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string Destination { get; set; } public int FailureTriggerTimeInterval { get; set; } public string FilterName { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string OnFatalErrorPolicy { get; set; } public string OtherOnFatalErrorPolicy { get; set; } public string OtherSubscriptionState { get; set; } public int RepeatNotificationCount { get; set; } public int RepeatNotificationGap { get; set; } public int RepeatNotificationInterval { get; set; } public string RepeatNotificationPolicy { get; set; } public string SubscriptionState { get; set; } }`

22 Indicator LED

Indicator LED status and control.

22.1 Indicator LED Show

Shows detailed status and configuration for a specific indicator LED.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t indicatorled [0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" indicatorled[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/IndicatorLEDCollection/IndicatorLED",
	"@odata.type" : "#IndicatorLED.v1_0_0.IndicatorLED",
	"ActivationState" : "Lit",
	"Availability" : "Not Configured",
	"Color" : "Red",
	"CommunicationStatus" : "Communication OK",
	"ControlMode" : "Automatic",
	"ControlPattern" : "Elliptic Curve",
	"DefaultActivationState" : "Lit",
	"DetailedStatus" : "No Additional Information",
	"ElementName" : "Indicator LED:0",
	"EnabledDefault" : "Enabled",
	"EnabledState" : "Enabled",
	"ErrorDescription" : "All errors are cleared",
	"HealthState" : "OK",
	"IdentifyingDescriptions" : "N/A",
	"IndicatedConditions" : "N/A",
	"InstanceNumber" : 0,
	"LocationIndicator" : "On",
	"Name" : "Indicator LED:0",
	"OperatingStatus" : "Completed",
	"OperationalStatus" : "N/A",
	"OtherColorDescription" : "This colour indicates success",
	"OtherIndicatedConditionDescription" : "This colour indicates success",
	"OtherSupportedColorDescriptions" : "N/A",
	"OtherSupportedIndicatedConditionDescriptions" : "N/A",
	"PowerManagementCapabilities" : "N/A",
	"PrimaryStatus" : "OK",
	"RequestedState" : "Enabled",
	"Status" : "Success",
	"StatusDescriptions" : "N/A",
	"StatusInfo" : "Enabled",
	"SupportedActivationStates" : "N/A",
	"SupportedColors" : "N/A",
	"SupportedControlModes" : "N/A",
	"SupportedControlPatterns" : "N/A",
	"TransitioningToState" : "Enabled"
	}

Output Class	`public class IndicatorLEDMember`
	`{`
	`public string odataid { get; set; }`
	`public string odatatype { get; set; }`
	`public string ActivationState { get; set; }`
	`public string Availability { get; set; }`
	`public string Color { get; set; }`
	`public string CommunicationStatus { get; set; }`
	`public string ControlMode { get; set; }`
	`public string ControlPattern { get; set; }`
	`public string DefaultActivationState { get; set; }`
	`public string DetailedStatus { get; set; }`

```
    public string ElementName { get; set; }
    public string EnabledDefault { get; set; }
    public string EnabledState { get; set; }
    public string ErrorDescription { get; set; }
    public string HealthState { get; set; }
    public string IdentifyingDescriptions { get; set; }
    public string IndicatedConditions { get; set; }
    public int InstanceNumber { get; set; }
    public string LocationIndicator { get; set; }
    public string Name { get; set; }
    public string OperatingStatus { get; set; }
    public string OperationalStatus { get; set; }
    public string OtherColorDescription { get; set; }
    public string OtherIndicatedConditionDescription { get; set; }
    public string OtherSupportedColorDescriptions { get; set; }
    public string OtherSupportedIndicatedConditionDescriptions { get;
set; }
    public string PowerManagementCapabilities { get; set; }
    public string PrimaryStatus { get; set; }
    public string RequestedState { get; set; }
    public string Status { get; set; }
    public string StatusDescriptions { get; set; }
    public string StatusInfo { get; set; }
    public string SupportedActivationStates { get; set; }
    public string SupportedColors { get; set; }
    public string SupportedControlModes { get; set; }
    public string SupportedControlPatterns { get; set; }
    public string TransitioningToState { get; set; }
}
```

23 IP Configuration

Purpose: IP network configuration management.

23.1 IP Configuration Show

Shows detailed IP configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t ipconfiguration[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" ipconfiguration[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/IPConfigurationCollection/IPConfiguration", "@odata.type" : "#IPConfiguration.v1_0_0.IPConfiguration", "AddressOrigin" : "N/A", "AvailableRequestedStates" : "N/A", "Caption" : "IP Network Connection Caption", "CommunicationStatus" : "Communication OK", "DHCPOptionsToUse" : "N/A", "DNSServerAddresses" : "N/A", "Description" : "IP Network Connection Description", "DetailedStatus" : "No Additional Information", "DomainName" : "N/A", "ElementName" : "IP Network Connection:0", "EnabledDefault" : "Enabled", "EnabledState" : "Enabled", "GatewayAddresses" : "N/A", "HealthState" : "OK", "ID" : "CIM_IPNetworkConnection", "IPAddresses" : "N/A", "IPv6SubnetPrefixLengths" : "N/A", "InstallDate" : "N/A", "InstanceID" : "IP Network Connection:0", "InstanceNumber" : 0, "OperatingStatus" : "In Service", "OperationalStatus" : "N/A", "OtherEnabledState" : "IP Network Connection:0", "PrimaryStatus" : "OK", "ProtocolIFType" : "N/A", "RequestedOptions" : "N/A", "RequestedState" : "Enabled", "RequiredOptions" : "N/A", "Status" : "Success", "StatusDescriptions" : "N/A", "SubnetMasks" : "N/A", "TimeOfLastStateChange" : "N/A", "TransitioningToState" : "N/A", "strName" : "IP Configuration" }`
Output Class	`public class IPConfigurationMember { public string odataid { get; set; } public string odatatype { get; set; } public string AddressOrigin { get; set; } public string AvailableRequestedStates { get; set; } public string Caption { get; set; } public string CommunicationStatus { get; set; } public string DHCPOptionsToUse { get; set; } public string DNSServerAddresses { get; set; } public string Description { get; set; } public string DetailedStatus { get; set; } public string DomainName { get; set; } public string ElementName { get; set; }`

```
    public string EnabledDefault { get; set; }
    public string EnabledState { get; set; }
    public string GatewayAddresses { get; set; }
    public string HealthState { get; set; }
    public string ID { get; set; }
    public string IPAddresses { get; set; }
    public string IPv6SubnetPrefixLengths { get; set; }
    public string InstallDate { get; set; }
    public string InstanceID { get; set; }
    public int InstanceNumber { get; set; }
    public string OperatingStatus { get; set; }
    public string OperationalStatus { get; set; }
    public string OtherEnabledState { get; set; }
    public string PrimaryStatus { get; set; }
    public string ProtocolIFType { get; set; }
    public string RequestedOptions { get; set; }
    public string RequestedState { get; set; }
    public string RequiredOptions { get; set; }
    public string Status { get; set; }
    public string StatusDescriptions { get; set; }
    public string SubnetMasks { get; set; }
    public string TimeOfLastStateChange { get; set; }
    public string TransitioningToState { get; set; }
    public string strName { get; set; }
}
```

24 IP Interface

Purpose: Ip information and other network data

24.1 IP Interface Show

Shows detailed IP interface for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t ipinterface[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" ipinterface[0]", "Commands":["show"]}`

Output JSON	<pre>{
	"@odata.id" : "/DASH/v1/IPInterfaceCollection/IPInterface",
	"@odata.type" : "#IPInterface.v1_0_0.IPInterface",
	"AddressOrigin" : "DHCP",
	"ElementName" : "IP Protocol:0",
	"EnabledState" : "Enabled",
	"IPName" : "IP Interface",
	"IPv4Address" : "127.0.0.1",
	"IPv6Address" : "N/A",
	"IPv6AddressType" : "N/A",
	"IPv6SubnetPrefixLength" : "N/A",
	"InstanceNumber" : 0,
	"Name" : "IPInterface",
	"NameFormat" : "2",
	"ProtocolIFType" : "IPv4",
	"RequestedState" : "Not Applicable",
	"SubnetMask" : "255.255.255.0"
	}</pre>

Output Class	<pre>public class IPInterfaceMember
	{
	public string odataid { get; set; }
	public string odatatype { get; set; }
	public string AddressOrigin { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string IPName { get; set; }
	public string IPv4Address { get; set; }
	public string IPv6Address { get; set; }
	public string IPv6AddressType { get; set; }
	public string IPv6SubnetPrefixLength { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public string NameFormat { get; set; }
	public string ProtocolIFType { get; set; }
	public string RequestedState { get; set; }
	public string SubnetMask { get; set; }
	}</pre>

25 KVM Redirection

Purpose: To debug and configure Remote system by Keyboard Video Mouse (KVM) redirection.

25.1 KVM Redirection Show

Shows detailed status of KVM Redirection for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t kvmredirection[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"kvmredirection[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/KVMRedirectionCollection/KVMRedirection", "@odata.type" : "#KVMRedirection.v1_0_0.KVMRedirection", "CreationClassName" : "CIM_KVMRedirectionSAP", "ElementName" : "KVMRedirectionSAP:0", "EnabledState" : "Disabled", "Id" : "KVMRedirection", "InstanceNumber" : 0, "KVMProtocol" : "RDP", "MaxCurrentEnabledSAPs" : "2", "Name" : "KVMRedirectionSAP", "RedirectionServiceType" : ["KVM"], "SharingMode" : "Exclusive" }`
Output Class	`public class KVMRedirectionMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string CreationClassName { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string Id { get; set; } public int InstanceNumber { get; set; } public string KVMProtocol { get; set; } public string MaxCurrentEnabledSAPs { get; set; } public string Name { get; set; } public List<string> RedirectionServiceType { get; set; } public string SharingMode { get; set; } }`

25.2 KVM Redirection Actions

Manages KVM Redirection states (show, enable, disable, connect, startkvm, startoskvm, startwinrekvm).

	<<option>>	
DASH CLI Command		`dashcli -h dash-system -u admin -P adminpass -t kvmredirection[0] <<option>>`
jdo Usage		`dashcli -jdo`
URI		`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON		`{"h":"dash-system","u":"admin","P":"adminpass","t":"kvmredirection[0]", "Commands":["<<option>>"]}`
Output JSON	enable	`{ "ErrorCode" : 0, "ErrorMessage" : "KVM SAP enabled successfully" }`
	disable	`{ "ErrorCode" : 0, "ErrorMessage" : "KVM SAP Disabled successfully" }`
	connect	`{ "ErrorCode" : 0,`

		"ErrorMessage" : "Launch KVM-VNC Viewer Succeeded" }

	startkvm	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Launch KVM-VNC Viewer Succeeded"
		}

	startoskvm	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Launch KVM-VNC Viewer Succeeded"
		}

	startwinrekvm	{
		"ErrorCode" : 0,
		"ErrorMessage" : "Launch KVM-VNC Viewer Succeeded"
		}

	stopwinrekvm	{ "ErrorCode" : 0, "ErrorMessage" : "Session stopped successfully" }

Output Class		public class CommandError {
		public int ErrorCode { get; set; }
		public string ErrorMessage { get; set; }
		}

26 Log Entry

Purpose: System log entry retrieval and management.

26.1 Log Entry Show

Shows detailed information for a specific log entry.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t logentry[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"logentry[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/LogEntryCollection/LogEntry", "@odata.type" : "#LogEntry.v1_0_0.LogEntry", "CreationTimeStamp" : "197001 1000000.000000-000", "ElementName" : "info Entry 256", "InstanceID" : "0", "InstanceNumber" : 0, "LogInstanceID" : "id", "LogName" : "info", "MessageArguments" : "N/A", "MessageID" : "N/A", "Name" : "LogEntry", "OwingEntity" : "N/A", "PerceivedSeverity" : "N/A", "RecordData" : "Reserved for future definition by this specification\n", "RecordFormat" : "RecordType*string", "RecordID" : "256" }`
Output Class	`public class LogEntryMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string CreationTimeStamp { get; set; } public string ElementName { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string LogInstanceID { get; set; } public string LogName { get; set; } public List<string> MessageArguments { get; set; } public string MessageID { get; set; } public string Name { get; set; } public string OwingEntity { get; set; } public int PerceivedSeverity { get; set; } public string RecordData { get; set; } public string RecordFormat { get; set; } public string RecordID { get; set; } }`

27 Media Redirection

Purpose: Virtual media redirection (CD/DVD, ISO) management.

27.1 Media Redirection Show

Shows details for a specific media redirection instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t mediaredirection[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" mediaredirection [0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/MediaRedirectionCollection", "@odata.type" : "#MediaRedirectionCollection.v1_0_0.MediaRedirectionCollection", "Id" : "MediaRedirectionCollection", "Members" : [{ "@odata.id" : "/DASH/v1/MediaRedirectionCollection/MediaRedirection", "@odata.type" : "#MediaRedirection.v1_0_0.MediaRedirection", "Caption" : "system:0", "CommandTimeout" : "N/A", "CommunicationStatus" : "Communication OK", "ConnectionMode" : "Connect", "Description" : "system:0", "DetailedStatus" : "Not Available", "ElementName" : "Media Redirection:0", "EnabledDefault" : "Enabled", "EnabledState" : "Enabled", "HealthState" : "OK", "InstallDate" : "N/A", "InstanceID" : "Media Redirection:0", "InstanceNumber" : 0, "Name" : "MediaRedirection", "OperatingStatus" : "Completed", "OtherEnabledState" : "system:0", "PrimaryStatus" : "OK", "RequestedState" : "Enabled", "ResetTimeout" : "N/A", "SessionTimeout" : "N/A", "Status" : "system:0", "TimeOfLastStateChange" : "N/A", "TransitioningToState" : "Test" }], "Name" : "Media Redirection Collection" }`
Output Class	`public class MediaRedirection { public string odataid { get; set; } public string odatatype { get; set; } public string Id { get; set; } public MediaRedirectionMember[] Members { get; set; } public string Name { get; set; } } public class MediaRedirectionMember { public string odataid { get; set; } public string odatatype { get; set; } public string Caption { get; set; } public string CommandTimeout { get; set; }`

```
    public string CommunicationStatus { get; set; }
    public string ConnectionMode { get; set; }
    public string Description { get; set; }
    public string DetailedStatus { get; set; }
    public string ElementName { get; set; }
    public string EnabledDefault { get; set; }
    public string EnabledState { get; set; }
    public string HealthState { get; set; }
    public string InstallDate { get; set; }
    public string InstanceID { get; set; }
    public int InstanceNumber { get; set; }
    public string Name { get; set; }
    public string OperatingStatus { get; set; }
    public string OtherEnabledState { get; set; }
    public string PrimaryStatus { get; set; }
    public string RequestedState { get; set; }
    public string ResetTimeout { get; set; }
    public string SessionTimeout { get; set; }
    public string Status { get; set; }
    public string TimeOfLastStateChange { get; set; }
    public string TransitioningToState { get; set; }
}
```

28 Memory

Purpose: Memory device status and information.

28.1 Memory Show

Shows detailed information for a specific memory device.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t memory[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"memory[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/LogEntryCollection/LogEntry", "@odata.type" : "#LogEntry.v1_0_0.LogEntry", "CreationTimeStamp" : "197001 1000000.000000-000", "ElementName" : "info Entry 256", "InstanceID" : "0", "InstanceNumber" : 0, "LogInstanceID" : "id", "LogName" : "info", "MessageArguments" : "N/A", "MessageID" : "N/A", "Name" : "LogEntry", "OwingEntity" : "N/A", "PerceivedSeverity" : "N/A", "RecordData" : "Reserved for future definition by this specification\n", "RecordFormat" : "RecordType*string", "RecordID" : "256" }`
Output Class	`public class MemoryMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string AccessType { get; set; } public string AvailableMemory { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string IsVolatileMemory { get; set; } public string Name { get; set; } public string RequestedState { get; set; } public string TotalMemory { get; set; } }`

28.2 Enumerate Asset

Lists assets associated with a specific memory device.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t memory[0] enumerate asset`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"memory[0]", "Commands":["enumerate","asset"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection", "@odata.type" : "#PhysicalAssetCollection.v1_0_0.PhysicalAssetCollection", "Id" : "PhysicalAssetCollection", "Members" : [{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset", "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset", "CanbeFRUed" : "No", "ElementName" : "Physical Memory Device #1",`

<table>
<tr><td></td><td>

```
            "InstanceNumber" : 0,
            "Manufacturer" : "JEDEC ID:80 CE",
            "MemoryBankLabel" : "SMBIOS_003d",
            "MemoryCapacity" : "2048 MB",
            "MemoryFormFactor" : "DIMM",
            "MemoryType" : "DDR3",
            "Model" : "N/A",
            "Name" : "PhysicalAsset",
            "PartNumber" : "M378B5673FH0-CH9",
            "SKU" : "N/A",
            "SerialNumber" : "822D9E63",
            "Tag" : "BRCM:5.1"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ElementName" : "Physical Memory Device #2",
            "InstanceNumber" : 1,
            "Manufacturer" : "JEDEC ID:80 CE",
            "MemoryBankLabel" : "SMBIOS_003e",
            "MemoryCapacity" : "2048 MB",
            "MemoryFormFactor" : "DIMM",
            "MemoryType" : "DDR3",
            "Model" : "N/A",
            "Name" : "PhysicalAsset",
            "PartNumber" : "M378B5673FH0-CH9",
            "SKU" : "N/A",
            "SerialNumber" : "7F2D9E63",
            "Tag" : "BRCM:5.2"
         }
      ],
      "Name" : "Physical Asset Collection"
}
```

</td></tr>
<tr><td>Output Class</td><td>

```
    public class PhysicalAssetCollection {
        [JsonProperty("@odata.id")]
        public string _odata_id { get; set; }
        [JsonProperty("@odata.type")]
        public string _odata_type { get; set; }
        public string Id { get; set; }
        [JsonProperty("Members")]
        public List<PhysicalAssetMember> Members { get; set; }
        public string Name { get; set; }
    }
    public class PhysicalAssetMember {
        [JsonProperty("@odata.id")]
        public string _odata_id { get; set; }
        [JsonProperty("@odata.type")]
        public string odatatype { get; set; }
        public string ConnectorLayout { get; set; }
        public List<string> ConnectorType { get; set; }
        public string ElementName { get; set; }
        public int InstanceNumber { get; set; }
        public string Manufacturer { get; set; }
        public string Model { get; set; }
        public string Name { get; set; }
        public string PartNumber { get; set; }
        public string SKU { get; set; }
        public string SerialNumber { get; set; }
        public int SlotNumber { get; set; }
        public string Tag { get; set; }
        public string CanbeFRUed { get; set; }
        public string MemoryBankLabel { get; set; }
        public string MemoryCapacity { get; set; }
        public string MemoryFormFactor { get; set; }
        public string MemoryType { get; set; }
        public string IsHostingBoard { get; set; }
        public string PackageType { get; set; }
```

</td></tr>
</table>

	public string ChassisType { get; set; } }

28.3 Asset Show

Shows a specific asset linked to a memory device.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t memory[0]/asset[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"memory[0]/asset[0]", "Commands":["show"]}

Output JSON	{
	"@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
	"@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
	"CanbeFRUed" : "No",
	"ElementName" : "Physical Memory Device #1",
	"InstanceNumber" : 0,
	"Manufacturer" : "JEDEC ID:80 CE",
	"MemoryBankLabel" : "SMBIOS_003d",
	"MemoryCapacity" : "2048 MB",
	"MemoryFormFactor" : "DIMM",
	"MemoryType" : "DDR3",
	"Model" : "N/A",
	"Name" : "PhysicalAsset",
	"PartNumber" : "M378B5673FH0-CH9",
	"SKU" : "N/A",
	"SerialNumber" : "822D9E63",
	"Tag" : "BRCM:5.1"
	}

Output Class	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }
	public int InstanceNumber { get; set; }
	public string Manufacturer { get; set; }
	public string Model { get; set; }
	public string Name { get; set; }
	public string PartNumber { get; set; }
	public string SKU { get; set; }
	public string SerialNumber { get; set; }
	public int SlotNumber { get; set; }
	public string Tag { get; set; }
	public string CanbeFRUed { get; set; }
	public string MemoryBankLabel { get; set; }
	public string MemoryCapacity { get; set; }
	public string MemoryFormFactor { get; set; }
	public string MemoryType { get; set; }
	public string IsHostingBoard { get; set; }
	public string PackageType { get; set; }
	public string ChassisType { get; set; }
	}

29 Network Port

Purpose: General network port configuration (includes Ethernet/Wi-Fi).

29.1 Network Port Show

Shows details for a specific network port.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t networkport[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"networkport[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id": "/DASH/v1/NetworkPortCollection/NetworkPort", "@odata.type": "#NetworkPort.v1_0_0.NetworkPort", "ChannelNumber": -1, "DeviceID": "26", "ElementName": "NetworkPort:0", "EnabledState": "Enabled", "InstanceNumber": 0, "LinkTechnology": "Ethernet", "MACAddress": "00:00:1a:20:6a:bf", "MaximumSpeed": "N/A", "Name": "Realtek PCIe GbE Family Controller", "RequestedSpeed": "N/A", "RequestedState": "Not Applicable", "Speed": "1 Gbps" }`
Output Class	`public class NetworkPortMember { public string odataid { get; set; } public string odatatype { get; set; } public int ChannelNumber { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string LinkTechnology { get; set; } public string MACAddress { get; set; } public string MaximumSpeed { get; set; } public string Name { get; set; } public string RequestedSpeed { get; set; } public string RequestedState { get; set; } public string Speed { get; set; } }`

30 Opaque Management Data

Purpose: Custom management data storage and retrieval.

30.1 Opaque Management Data Show

Shows details for a specific opaque management data object.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t opaquemanagementdata [0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" opaquemanagementdata [0]","Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/OpaqueManagementDataCollection/OpaqueManagementData",
	"@odata.type" : "#OpaqueManagementData.v1_0_0.OpaqueManagementData",
	"Access" : "Read/Write Supported",
	"BlockSize" : 300,
	"ConsumableBlocks" : 20,
	"DataOrganization" : "Variable Block",
	"DeviceID" : "Opaque Management Data Device: 0",
	"ElementName" : "Opaque Management Data:0",
	"InstanceNumber" : 0,
	"LastAccessed" : "N/A",
	"MaxSize" : 13,
	"Name" : "OpaqueManagementData",
	"NumberOfBlocks" : 30,
	"OwnerName" : "N/A",
	"WriteLimited" : "Write-limited"
	}

Output Class	`public class OpaqueManagementDataMember`
	`{`
	`public string odataid { get; set; }`
	`public string odatatype { get; set; }`
	`public string Access { get; set; }`
	`public int BlockSize { get; set; }`
	`public int ConsumableBlocks { get; set; }`
	`public string DataOrganization { get; set; }`
	`public string DeviceID { get; set; }`
	`public string ElementName { get; set; }`
	`public int InstanceNumber { get; set; }`
	`public string LastAccessed { get; set; }`
	`public int MaxSize { get; set; }`
	`public string Name { get; set; }`
	`public int NumberOfBlocks { get; set; }`
	`public string OwnerName { get; set; }`
	`public string WriteLimited { get; set; }`
	`}`

31 Operating System

Purpose: Operating system information management.

31.1 Operating System Show

Shows detailed OS information for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t operatingsystem[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"operatingsystem[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/OperatingSystemCollection/OperatingSystem", "@odata.type" : "#OperatingSystem.v1_0_0.OperatingSystem", "AvailableRequestedStates" : "N/A", "EnabledState" : "Enabled", "InstanceNumber" : 0, "Name" : "Microsoft Windows 10 Enterprise", "OSType" : "Microsoft Windows 10 64-bit", "RequestedState" : "No Change", "TransitioningToState" : "N/A" }`
Output Class	`public class OperatingSystemMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public List<string> AvailableRequestedStates { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string OSType { get; set; } public string RequestedState { get; set; } public string TransitioningToState { get; set; } }`

32 PCI Device

Purpose: PCI device enumeration and information.

32.1 PCI Device Show

Shows detailed information for a specific PCI device.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t pcidevice[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" pcidevice[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/PCIDeviceCollection/PCIDevice",
	"@odata.type" : "#PCIDevice.v1_0_0.PCIDevice",
	"AdditionalAvailability" : "N/A",
	"AllocationState" : "N/A",
	"Availability" : "Running/Full Power",
	"AvailableRequestedStates" : "N/A",
	"BaseAddress" : "N/A",
	"Capabilities" : "N/A",
	"CapabilityDescriptions" : "N/A",
	"Caption" : "Default Value",
	"CommunicationStatus" : "Communication OK",
	"Description" : "Default Value",
	"DetailedStatus" : "Stressed",
	"DeviceID" : "PCI Device:0",
	"ElementName" : "PCI Device:0",
	"EnabledDefault" : "Enabled",
	"EnabledState" : "Enabled",
	"ErrorDescription" : "Danger",
	"HealthState" : "OK",
	"IdentifyingDescriptions" : "N/A",
	"InstanceID" : "PCI Device:0",
	"InstanceNumber" : 0,
	"LocationIndicator" : "On",
	"Name" : "PCIDevice",
	"OperatingStatus" : "Completed",
	"OperationalStatus" : "N/A",
	"OtherEnabledState" : "Default",
	"OtherIdentifyingInfo" : "N/A",
	"PowerManagementCapabilities" : "N/A",
	"PrimaryStatus" : "OK",
	"ProtocolDescription" : "Default",
	"ProtocolSupported" : "I2C",
	"RequestedState" : "Enabled",
	"Status" : "Success",
	"StatusDescriptions" : "N/A",
	"StatusInfo" : "Enabled",
	"SystemName" : "GURU-WIN11",
	"TransitioningToState" : "Enabled"
	}

Output Class	`public class PCIDeviceMember`
	`{`
	`public string odataid { get; set; }`
	`public string odatatype { get; set; }`
	`public string AdditionalAvailability { get; set; }`
	`public string AllocationState { get; set; }`
	`public string Availability { get; set; }`
	`public string AvailableRequestedStates { get; set; }`
	`public string BaseAddress { get; set; }`
	`public string Capabilities { get; set; }`
	`public string CapabilityDescriptions { get; set; }`
	`public string Caption { get; set; }`
	`public string CommunicationStatus { get; set; }`

```
    public string Description { get; set; }
    public string DetailedStatus { get; set; }
    public string DeviceID { get; set; }
    public string ElementName { get; set; }
    public string EnabledDefault { get; set; }
    public string EnabledState { get; set; }
    public string ErrorDescription { get; set; }
    public string HealthState { get; set; }
    public string IdentifyingDescriptions { get; set; }
    public string InstanceID { get; set; }
    public int InstanceNumber { get; set; }
    public string LocationIndicator { get; set; }
    public string Name { get; set; }
    public string OperatingStatus { get; set; }
    public string OperationalStatus { get; set; }
    public string OtherEnabledState { get; set; }
    public string OtherIdentifyingInfo { get; set; }
    public string PowerManagementCapabilities { get; set; }
    public string PrimaryStatus { get; set; }
    public string ProtocolDescription { get; set; }
    public string ProtocolSupported { get; set; }
    public string RequestedState { get; set; }
    public string Status { get; set; }
    public string StatusDescriptions { get; set; }
    public string StatusInfo { get; set; }
    public string SystemName { get; set; }
    public string TransitioningToState { get; set; }
}
```

33 Physical Computer System View

Purpose: Physical View of Computer System and related profiles.

33.1 Physical Computer System View Show

Shows detailed physical system view for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t physicalcomputersystemview [0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"physicalcomputersystemview [0]", "Commands":["show"]}`
Output JSON	{ "@odata.id" : "/DASH/v1/PhysicalComputerSystemViewCollection/PhysicalComputerSystemView", "@odata.type" : "#PhysicalComputerSystemView.v1_0_0.PhysicalComputerSystemView", "CurrentBIOSBuildNumber" : 0, "CurrentBIOSMajorVersion" : 1, "CurrentBIOSMinorVersion" : 0, "CurrentBIOSRevisionNumber" : 0, "CurrentBIOSVersionString" : "AMD BIOS:1.0.0.0", "CurrentManagementFirmwareBuildNumber" : 0, "CurrentManagementFirmwareElementName" : "AMD Management Firmware", "CurrentManagementFirmwareMajorVersion" : 1, "CurrentManagementFirmwareMinorVersion" : 0, "CurrentManagementFirmwareRevisionNumber" : 0, "CurrentManagementFirmwareVersionString" : "AMD Management Firmware:1.0.0.0", "Dedicated" : "N/A", "EnabledState" : "Enabled", "FRUInfoSupported" : "Yes", "HealthState" : "OK", "IdentifyingDescriptions" : "N/A", "InstanceID" : "Physical Computer System View:0", "InstanceNumber" : 0, "LogCurrentNumberOfRecords" : "N/A", "LogInstanceID" : "N/A", "LogMaxNumberOfRecords" : "N/A", "LogOverWritePolicy" : "N/A", "LogState" : "N/A", "Manufacturer" : "AMD", "MemoryBlockSize" : 400, "MemoryConsumableBlocks" : 56765, "MemoryNumberOfBlocks" : 65000, "Model" : "Model:2020", "Name" : "PhysicalComputerSystemView", "NumberOfProcessorCores" : 1, "NumberOfProcessorThreads" : 1, "NumberOfProcessors" : 1, "NumericSensorBaseUnits" : "N/A", "NumericSensorContext" : "N/A", "NumericSensorCurrentReading" : "N/A", "NumericSensorCurrentState" : "N/A", "NumericSensorElementName" : "N/A", "NumericSensorEnabledState" : "N/A", "NumericSensorHealthState" : "N/A", "NumericSensorLowerThresholdCritical" : "N/A", "NumericSensorLowerThresholdFatal" : "N/A", "NumericSensorLowerThresholdNonCritical" : "N/A", "NumericSensorOtherSensorTypeDescription" : "N/A", "NumericSensorPrimaryStatus" : "N/A", "NumericSensorRateUnits" : "N/A", "NumericSensorSensorType" : "N/A", "NumericSensorUnitModifier" : "N/A", "NumericSensorUpperThresholdCritical" : "N/A",

<table>
<tr><td></td><td><pre>
        "NumericSensorUpperThresholdFatal" : "N/A",
        "NumericSensorUpperThresholdNonCritical" : "N/A",
        "OSEnabledState" : "Enabled",
        "OSType" : "Microsoft Windows 10 64-bit",
        "OSVersion" : "Microsoft Windows 10 64-bit",
        "OneTimeBootSource" : 1,
        "OperationalStatus" : "N/A",
        "OtherDedicatedDescriptions" : "N/A",
        "OtherIdentifyingInfo" : "N/A",
        "PartNumber" : "PartNumber:2020",
        "PersistentBootConfigOrder" : "N/A",
        "PowerAllocationLimit" : 1,
        "PowerUtilizationMode" : "Dynamic",
        "PowerUtilizationModesSupported" : "N/A",
        "ProcessorCurrentClockSpeed" : 1000,
        "ProcessorFamily" : "AMD FirePro(TM) Series Processor",
        "ProcessorMaxClockSpeed" : 2500,
        "RequestedState" : "Enabled",
        "SKU" : "SKU:2020",
        "SerialNumber" : "SerialNumber:2020",
        "StructuredBootString" : "N/A",
        "Tag" : "Physical Computer System View Tag",
        "Version" : "Version:2020"
}
</pre></td></tr>
<tr><td>Output Class</td><td><pre>
public class PhysicalComputerSystemView
{
    public string odataid { get; set; }
    public string odatatype { get; set; }
    public int CurrentBIOSBuildNumber { get; set; }
    public int CurrentBIOSMajorVersion { get; set; }
    public int CurrentBIOSMinorVersion { get; set; }
    public int CurrentBIOSRevisionNumber { get; set; }
    public string CurrentBIOSVersionString { get; set; }
    public int CurrentManagementFirmwareBuildNumber { get; set; }
    public string CurrentManagementFirmwareElementName { get; set; }
    public int CurrentManagementFirmwareMajorVersion { get; set; }
    public int CurrentManagementFirmwareMinorVersion { get; set; }
    public int CurrentManagementFirmwareRevisionNumber { get; set; }
    public string CurrentManagementFirmwareVersionString { get; set; }
    public string Dedicated { get; set; }
    public string EnabledState { get; set; }
    public string FRUInfoSupported { get; set; }
    public string HealthState { get; set; }
    public string IdentifyingDescriptions { get; set; }
    public string InstanceID { get; set; }
    public int InstanceNumber { get; set; }
    public string LogCurrentNumberOfRecords { get; set; }
    public string LogInstanceID { get; set; }
    public string LogMaxNumberOfRecords { get; set; }
    public string LogOverWritePolicy { get; set; }
    public string LogState { get; set; }
    public string Manufacturer { get; set; }
    public int MemoryBlockSize { get; set; }
    public int MemoryConsumableBlocks { get; set; }
    public int MemoryNumberOfBlocks { get; set; }
    public string Model { get; set; }
    public string Name { get; set; }
    public int NumberOfProcessorCores { get; set; }
    public int NumberOfProcessorThreads { get; set; }
    public int NumberOfProcessors { get; set; }
    public string NumericSensorBaseUnits { get; set; }
</pre></td></tr>
</table>

```
    public string NumericSensorContext { get; set; }
    public string NumericSensorCurrentReading { get; set; }
    public string NumericSensorCurrentState { get; set; }
    public string NumericSensorElementName { get; set; }
    public string NumericSensorEnabledState { get; set; }
    public string NumericSensorHealthState { get; set; }
    public string NumericSensorLowerThresholdCritical { get; set; }
    public string NumericSensorLowerThresholdFatal { get; set; }
    public string NumericSensorLowerThresholdNonCritical { get; set; }
    public string NumericSensorOtherSensorTypeDescription { get; set; }
    public string NumericSensorPrimaryStatus { get; set; }
    public string NumericSensorRateUnits { get; set; }
    public string NumericSensorSensorType { get; set; }
    public string NumericSensorUnitModifier { get; set; }
    public string NumericSensorUpperThresholdCritical { get; set; }
    public string NumericSensorUpperThresholdFatal { get; set; }
    public string NumericSensorUpperThresholdNonCritical { get; set; }
    public string OSEnabledState { get; set; }
    public string OSType { get; set; }
    public string OSVersion { get; set; }
    public int OneTimeBootSource { get; set; }
    public string OperationalStatus { get; set; }
    public string OtherDedicatedDescriptions { get; set; }
    public string OtherIdentifyingInfo { get; set; }
    public string PartNumber { get; set; }
    public string PersistentBootConfigOrder { get; set; }
    public int PowerAllocationLimit { get; set; }
    public string PowerUtilizationMode { get; set; }
    public string PowerUtilizationModesSupported { get; set; }
    public int ProcessorCurrentClockSpeed { get; set; }
    public string ProcessorFamily { get; set; }
    public int ProcessorMaxClockSpeed { get; set; }
    public string RequestedState { get; set; }
    public string SKU { get; set; }
    public string SerialNumber { get; set; }
    public string StructuredBootString { get; set; }
    public string Tag { get; set; }
    public string Version { get; set; }
}
```

34 Platform Watchdog

Purpose: Watchdog timer configuration and status.

34.1 Platform Watchdog Show

Shows the current status and settings of the platform watchdog.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t platformwatchdog [0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" platformwatchdog[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/PlatformWatchdogCollection/PlatformWatchdog",
	"@odata.type" : "#PlatformWatchdog.v1_0_0.PlatformWatchdog",
	"ActionOnExpiration" : "Reset Monitored Entity",
	"AvailableRequestedStates" : "N/A",
	"CommunicationStatus" : "Communication OK",
	"DetailedStatus" : "Not Available",
	"ElementName" : "Platform Watchdog:0",
	"EnabledDefault" : "Enabled",
	"EnabledState" : "Enabled",
	"HealthState" : "OK",
	"InstanceNumber" : 0,
	"LoSID" : "Watchdog Active",
	"LoSOrgID" : "AMD",
	"MonitoredEntityType" : "Operating System",
	"Name" : "Watchdog",
	"OperatingStatus" : "Completed",
	"OperationalStatus" : "N/A",
	"OtherActionOnExpiration" : "Restart System",
	"OtherEnabledState" : "None",
	"PrimaryOwnerContact" : "AMD",
	"PrimaryOwnerName" : "DASH Admin",
	"PrimaryStatus" : "OK",
	"RequestedState" : "Enabled",
	"StartMode" : "Kick start",
	"Status" : "OK",
	"StatusDescriptions" : "N/A",
	"TransitioningToState" : "Enabled"
	}

Output Class	public class Rootobject { public string odataid { get; set; } public string odatatype { get; set; } public string ActionOnExpiration { get; set; } public string AvailableRequestedStates { get; set; } public string CommunicationStatus { get; set; } public string DetailedStatus { get; set; } public string ElementName { get; set; } public string EnabledDefault { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string LoSID { get; set; } public string LoSOrgID { get; set; } public string MonitoredEntityType { get; set; } public string Name { get; set; } public string OperatingStatus { get; set; } public string OperationalStatus { get; set; } public string OtherActionOnExpiration { get; set; } public string OtherEnabledState { get; set; } public string PrimaryOwnerContact { get; set; }

```
    public string PrimaryOwnerName { get; set; }
    public string PrimaryStatus { get; set; }
    public string RequestedState { get; set; }
    public string StartMode { get; set; }
    public string Status { get; set; }
    public string StatusDescriptions { get; set; }
    public string TransitioningToState { get; set; }
}
```

35 Power Scheduler

Purpose: To schedule power operations at a predetermined future time.

35.1 Power Scheduler Show

Shows detailed information for a specific power scheduler instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t powerscheduler[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"powerscheduler[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PowerSchedulerCollection/PowerScheduler", "@odata.type" : "#PowerScheduler.v1_0_0.PowerScheduler", "CreationClassName" : "AIMT_PowerScheduler", "Datetime" : "00000000104500.000000+000", "ElementName" : "AIMT_PowerScheduler.1", "InstanceID" : "AIMT:01.1", "InstanceNumber" : 0, "Name" : "PowerOn1045", "SystemCreationClassName" : "CIM_ComputerSystem", "SystemName" : "dash-system" }`
Output Class	`public class Rootobject { public string odataid { get; set; } public string odatatype { get; set; } public string CreationClassName { get; set; } public string Datetime { get; set; } public string ElementName { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string SystemCreationClassName { get; set; } public string SystemName { get; set; } }`

35.2 Power Scheduler Add

Adds an entry for a power scheduler instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass powerscheduler add 08:36:36 2 Start836`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":" dash-system","u":" admin ","P":" adminpass ","C":1,"Commands":["powerscheduler","add","10:45:00","2","PowerOn1045"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Powerscheduler created successfully", "VerboseMessage" : "" }`
Output Class	`public class Rootobject { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } public string VerboseMessage { get; set; } }`

35.3 Power Scheduler Clear

Clears all the entries of power scheduler.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass powerscheduler clear`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":" dash-system","u":" admin ","P":" adminpass ","C":1,"Commands":["powerscheduler","clear"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Cleared all the instance successfully", "VerboseMessage" : "" }`
Output Class	`public class Rootobject { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } public string VerboseMessage { get; set; } }`

Power Scheduler Update

Updates an existing power scheduler entry on the target system.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass powerscheduler update
jdo Usage	dashcli -jdo {"h":"dash-system","u":"admin","P":"adminpass","t":"powerscheduler[0]", "Commands":["update", "03:23:10", 2, "Name"]}
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"powerscheduler[0]", "Commands":["update", "03:23:10", 2, "Name"]}
Output JSON	{ "ErrorCode" : 0, "ErrorMessage" : "Powerscheduler updated Successfully" }
Output Class	public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }

Power Scheduler Delete

Deletes a specific power scheduler entry from the target system.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t powerscheduler[0] delete
jdo Usage	dashcli -jdo {"h":"dash-system","u":"admin","P":"adminpass","t":"powerscheduler[0]", "Commands":["delete"]}
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"powerscheduler[0]", "Commands":["delete"]}
Output JSON	{ "ErrorCode" : 0, "ErrorMessage" : "Powerscheduler deleted Successfully" }
Output Class	public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }

36 Power Supply

Purpose: Power supply unit status and information.

36.1 Power Supply Show

Shows detailed information for a specific power supply.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t powersupply[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"powersupply[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PowerSupplyCollection/PowerSupply", "@odata.type" : "#PowerSupply.v1_0_0.PowerSupply", "DeviceID" : "BRCM:9.1", "ElementName" : "Power Supply #1", "EnabledState" : "Enabled", "HealthState" : "Unknown", "InstanceNumber" : 0, "Name" : "Power Supply", "OperationalState" : ["Unknown"], "RequestedState" : "Not Applicable", "TotalPower" : 0 }`
Output Class	`public class PowerSupplyMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public List<string> OperationalState { get; set; } public string RequestedState { get; set; } public int TotalPower { get; set; } }`

36.2 Enumerate Asset

Lists assets associated with a specific power supply.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t powersupply[0] enumerate asset`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"powersupply[0]", "Commands":["enumerate","asset"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection", "@odata.type" : "#PhysicalAssetCollection.v1_0_0.PhysicalAssetCollection", "Id" : "PhysicalAssetCollection", "Members" : [{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset", "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset", "CanbeFRUed" : "Yes", "ElementName" : "BaseBoard:0", "InstanceNumber" : 0, "IsHostingBoard" : "Yes", "Manufacturer" : "LENOVO", "Model" : ".MFG_IN_GO", "Name" : "PhysicalAsset", "PartNumber" : "Not Defined", "SKU" : "NULL",`

```
            "SerialNumber" : "L1HF8B1000S",
            "Tag" : "Card:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ChassisType" : "Desktop",
            "ElementName" : "Chassis:0",
            "InstanceNumber" : 1,
            "Manufacturer" : "LENOVO",
            "Model" : ".MFG_IN_GO",
            "Name" : "PhysicalAsset",
            "PackageType" : "Chassis/Frame",
            "PartNumber" : "None",
            "SKU" : "LENOVO_MT_.MFG_BU_Think_FM_ThinkPad T495",
            "SerialNumber" : "ING.",
            "Tag" : "Chassis:0",
            "Version" : "None"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ElementName" : "Processor:0",
            "InstanceNumber" : 2,
            "Manufacturer" : "Advanced Micro Devices, Inc.",
            "Model" : "AMD Eng Sample: ZM350SC4T4MFG_36/20_Y          ",
            "Name" : "PhysicalAsset",
            "PartNumber" : "None",
            "SKU" : "NULL",
            "SerialNumber" : "None",
            "Tag" : "Processor:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "CanbeFRUed" : "No",
            "ElementName" : "Memory:0",
            "InstanceNumber" : 3,
            "Manufacturer" : "Samsung",
            "MemoryBankLabel" : "P0 CHANNEL A",
            "MemoryCapacity" : "8192 MB",
            "MemoryFormFactor" : "DIMM",
            "MemorySpeed" : "2400 nanosecond",
            "MemoryType" : "DDR4",
            "Model" : "N/A",
            "Name" : "PhysicalAsset",
            "PartNumber" : "M471A1G44AB0-CTD    ",
            "SKU" : "N/A",
            "SerialNumber" : "00000000",
            "Tag" : "Mem:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:0",
            "InstanceNumber" : 4,
            "Manufacturer" : "N/A",
            "Model" : "PCI-E x1 (J3600)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:0"
         },
         {
```

```
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:1",
            "InstanceNumber" : 5,
            "Manufacturer" : "N/A",
            "Model" : "PCI-E x8 (J3605)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:1"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:2",
            "InstanceNumber" : 6,
            "Manufacturer" : "N/A",
            "Model" : "M.2 PCI-E Connector (J3700)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:2"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:3",
            "InstanceNumber" : 7,
            "Manufacturer" : "N/A",
            "Model" : "M.2 WWAN Connector (J3702)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:3"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:4",
            "InstanceNumber" : 8,
            "Manufacturer" : "N/A",
            "Model" : "M.2 WLAN/BT Connector (J3703)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:4"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:5",
            "InstanceNumber" : 9,
            "Manufacturer" : "N/A",
            "Model" : "PCIE SD_Card (J4000)",
            "Name" : "PhysicalAsset",
```

	"PartNumber" : "N/A", "SKU" : "N/A", "SerialNumber" : "N/A", "Tag" : "Slot:5" }], "Name" : "Physical Asset Collection" }

Output Class	public class PhysicalAssetCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	[JsonProperty("Members")]
	public List<PhysicalAssetMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }
	public int InstanceNumber { get; set; }
	public string Manufacturer { get; set; }
	public string Model { get; set; }
	public string Name { get; set; }
	public string PartNumber { get; set; }
	public string SKU { get; set; }
	public string SerialNumber { get; set; }
	public int SlotNumber { get; set; }
	public string Tag { get; set; }
	public string CanbeFRUed { get; set; }
	public string MemoryBankLabel { get; set; }
	public string MemoryCapacity { get; set; }
	public string MemoryFormFactor { get; set; }
	public string MemoryType { get; set; }
	public string IsHostingBoard { get; set; }
	public string PackageType { get; set; }
	public string ChassisType { get; set; }
	}

36.3 Asset

Shows a specific asset linked to a power supply.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t powersupply[0]/asset[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"powersupply[0]/asset[0]", "Commands":["show"]}`
Output JSON	{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset", "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset", "CanbeFRUed" : "Yes", "ElementName" : "BaseBoard:0", "InstanceNumber" : 0, "IsHostingBoard" : "Yes", "Manufacturer" : "LENOVO", "Model" : ".MFG_IN_GO", "Name" : "PhysicalAsset", "PartNumber" : "Not Defined", "SKU" : "NULL", "SerialNumber" : "L1HF8B1000S",

	"Tag" : "Card:0" }

Output Class	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }
	public int InstanceNumber { get; set; }
	public string Manufacturer { get; set; }
	public string Model { get; set; }
	public string Name { get; set; }
	public string PartNumber { get; set; }
	public string SKU { get; set; }
	public string SerialNumber { get; set; }
	public int SlotNumber { get; set; }
	public string Tag { get; set; }
	public string CanbeFRUed { get; set; }
	public string MemoryBankLabel { get; set; }
	public string MemoryCapacity { get; set; }
	public string MemoryFormFactor { get; set; }
	public string MemoryType { get; set; }
	public string IsHostingBoard { get; set; }
	public string PackageType { get; set; }
	public string ChassisType { get; set; }
	}

37 Processor

Purpose: Processor status and information.

37.1 Processor Show

Shows detailed information for a specific processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/ProcessorCollection/Processor", "@odata.type" : "#Processor.v1_0_0.Processor", "CPUStatus" : "CPU Enabled", "CurClockSpeed" : "2400 MHz", "DeviceID" : "AMD:20.0", "ElementName" : "Processor:0", "EnabledState" : "Enabled", "ExtBusSpeed" : "100 MHz", "Family" : "AMD Zen(TM) Processor Family", "HealthState" : "OK", "InstanceNumber" : 0, "LoadPercent" : "6%", "MaxClockSpeed" : "3850 MHz", "Name" : "Processor", "NumberOfCores" : 10, "NumberOfEnabledCores" : 10, "NumberOfThreadsPerProcessor" : 20, "OperationalState" : ["OK"], "RequestedState" : "Not Applicable", "Stepping" : "AMD Eng Sample: 100-000001713-21_N" }`
Output Class	`public class ProcessorMember { [JsonProperty("@odata.id")]`

```
        public string _odata_id { get; set; }
        [JsonProperty("@odata.type")]
        public string odatatype { get; set; }
        public string CPUStatus { get; set; }
        public string CurClockSpeed { get; set; }
        public string DeviceID { get; set; }
        public string ElementName { get; set; }
        public string EnabledState { get; set; }
        public string ExtBusSpeed { get; set; }
        public string Family { get; set; }
        public string HealthState { get; set; }
        public int InstanceNumber { get; set; }
        public string LoadPercent { get; set; }
        public string MaxClockSpeed { get; set; }
        public string Name { get; set; }
        public int NumberOfCores { get; set; }
        public int NumberOfEnabledCores { get; set; }
        public int NumberOfThreadsPerProcessor { get; set; }
        public List<string> OperationalState { get; set; }
        public string RequestedState { get; set; }
        public string Stepping { get; set; }
    }
```

37.2 Enumerate Asset

Lists assets associated with a specific processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0] enumerate asset`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]", "Commands":["enumerate","asset"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/PhysicalAssetCollection", "@odata.type" : "#PhysicalAssetCollection.v1_0_0.PhysicalAssetCollection", "Id" : "PhysicalAssetCollection", "Members" : [`

```
{
   "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
   "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
   "CanbeFRUed" : "Yes",
   "ElementName" : "BaseBoard:0",
   "InstanceNumber" : 0,
   "IsHostingBoard" : "Yes",
   "Manufacturer" : "LENOVO",
   "Model" : ".MFG_IN_GO",
   "Name" : "PhysicalAsset",
   "PartNumber" : "Not Defined",
   "SKU" : "NULL",
   "SerialNumber" : "L1HF8B1000S",
   "Tag" : "Card:0"
},
{
   "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
   "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
   "CanbeFRUed" : "No",
   "ChassisType" : "Desktop",
   "ElementName" : "Chassis:0",
   "InstanceNumber" : 1,
   "Manufacturer" : "LENOVO",
   "Model" : ".MFG_IN_GO",
   "Name" : "PhysicalAsset",
   "PackageType" : "Chassis/Frame",
   "PartNumber" : "None",
   "SKU" : "LENOVO_MT_.MFG_BU_Think_FM_ThinkPad T495",
   "SerialNumber" : "ING.",
   "Tag" : "Chassis:0",
   "Version" : "None"
},
{
   "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
   "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
   "CanbeFRUed" : "No",
   "ElementName" : "Processor:0",
   "InstanceNumber" : 2,
   "Manufacturer" : "Advanced Micro Devices, Inc.",
   "Model" : "AMD Eng Sample: ZM350SC4T4MFG_36/20_Y          ",
   "Name" : "PhysicalAsset",
   "PartNumber" : "None",
   "SKU" : "NULL",
   "SerialNumber" : "None",
   "Tag" : "Processor:0"
},
{
   "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
   "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
   "CanbeFRUed" : "No",
   "ElementName" : "Memory:0",
   "InstanceNumber" : 3,
   "Manufacturer" : "Samsung",
   "MemoryBankLabel" : "P0 CHANNEL A",
   "MemoryCapacity" : "8192 MB",
   "MemoryFormFactor" : "DIMM",
   "MemorySpeed" : "2400 nanosecond",
   "MemoryType" : "DDR4",
   "Model" : "N/A",
   "Name" : "PhysicalAsset",
   "PartNumber" : "M471A1G44AB0-CTD    ",
   "SKU" : "N/A",
   "SerialNumber" : "00000000",
   "Tag" : "Mem:0"
},
{
   "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
   "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
   "ConnectorLayout" : "Slot",
```

```
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:0",
            "InstanceNumber" : 4,
            "Manufacturer" : "N/A",
            "Model" : "PCI-E x1 (J3600)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:0"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:1",
            "InstanceNumber" : 5,
            "Manufacturer" : "N/A",
            "Model" : "PCI-E x8 (J3605)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:1"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:2",
            "InstanceNumber" : 6,
            "Manufacturer" : "N/A",
            "Model" : "M.2 PCI-E Connector (J3700)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:2"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:3",
            "InstanceNumber" : 7,
            "Manufacturer" : "N/A",
            "Model" : "M.2 WWAN Connector (J3702)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
            "Tag" : "Slot:3"
         },
         {
            "@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
            "@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
            "ConnectorLayout" : "Slot",
            "ConnectorType" : [ "Other" ],
            "ElementName" : "Slot:4",
            "InstanceNumber" : 8,
            "Manufacturer" : "N/A",
            "Model" : "M.2 WLAN/BT Connector (J3703)",
            "Name" : "PhysicalAsset",
            "PartNumber" : "N/A",
            "SKU" : "N/A",
            "SerialNumber" : "N/A",
```

	"Tag" : "Slot:4"
	},
	{
	"@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",
	"@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",
	"ConnectorLayout" : "Slot",
	"ConnectorType" : ["Other"],
	"ElementName" : "Slot:5",
	"InstanceNumber" : 9,
	"Manufacturer" : "N/A",
	"Model" : "PCIE SD_Card (J4000)",
	"Name" : "PhysicalAsset",
	"PartNumber" : "N/A",
	"SKU" : "N/A",
	"SerialNumber" : "N/A",
	"Tag" : "Slot:5"
	}
	],
	"Name" : "Physical Asset Collection"
	}

Output Class	```
	public class PhysicalAssetCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	[JsonProperty("Members")]
	public List<PhysicalAssetMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class PhysicalAssetMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string ConnectorLayout { get; set; }
	public List<string> ConnectorType { get; set; }
	public string ElementName { get; set; }
	public int InstanceNumber { get; set; }
	public string Manufacturer { get; set; }
	public string Model { get; set; }
	public string Name { get; set; }
	public string PartNumber { get; set; }
	public string SKU { get; set; }
	public string SerialNumber { get; set; }
	public int SlotNumber { get; set; }
	public string Tag { get; set; }
	public string CanbeFRUed { get; set; }
	public string MemoryBankLabel { get; set; }
	public string MemoryCapacity { get; set; }
	public string MemoryFormFactor { get; set; }
	public string MemoryType { get; set; }
	public string IsHostingBoard { get; set; }
	public string PackageType { get; set; }
	public string ChassisType { get; set; }
	}
	``

37.3 Enumerate Fan

Lists fans associated with a specific processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0] enumerate fan`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]", "Commands":["enumerate","fan"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/FanCollection",
	"@odata.type" : "#FanCollection.v1_0_0.FanCollection",
	"Id" : "FanCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/FanCollection/Fan",
	"@odata.type" : "#Fan.v1_0_0.Fan",
	"DesiredSpeed" : "N/A",
	"DeviceID" : "Fan:0",
	"ElementName" : "FAN_SPEED",
	"EnabledState" : "Enabled",
	"HealthState" : "OK",
	"InstanceNumber" : 0,
	"IsActiveCooling" : "Yes",
	"IsVariableSpeed" : "No",
	"Name" : "Fan",
	"OperationalState" : ["OK"],
	"RequestedState" : "Not Applicable"
	}
	],
	"Name" : "Fan Collection"
	}

Output Class	public class FanCollection {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string _odata_type { get; set; }
	public string Id { get; set; }
	[JsonProperty("Members")]
	public List<FanMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class FanMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public int DesiredSpeed { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string IsActiveCooling { get; set; }
	public string IsVariableSpeed { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public string RequestedState { get; set; }
	}

37.4 Enumerate Memory

Lists memory devices associated with a specific processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0] enumerate memory`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]", "Commands":["enumerate","memory"]}
Output JSON	{ "@odata.id" : "/DASH/v1/MemoryCollection", "@odata.type" : "#MemoryCollection.v1_0_0.MemoryCollection", "Id" : "MemoryCollection", "Members" : [{ "@odata.id" : "/DASH/v1/MemoryCollection/Memory", "@odata.type" : "#Memory.v1_0_0.Memory",

<table>
<tr><td></td><td>

<pre>
            "AccessType" : "Read/Write Supported",
            "AvailableMemory" : "8192 MB",
            "DeviceID" : "Memory:0",
            "ElementName" : "Total System Memory:0",
            "EnabledState" : "Enabled",
            "HealthState" : "OK",
            "InstanceNumber" : 0,
            "IsVolatileMemory" : "Yes",
            "Name" : "Memory",
            "RequestedState" : "Not Applicable",
            "TotalMemory" : "8192 MB"
        },
        {
            "@odata.id" : "/DASH/v1/MemoryCollection/Memory",
            "@odata.type" : "#Memory.v1_0_0.Memory",
            "AccessType" : "Read/Write Supported",
            "AvailableMemory" : "393216 Bytes",
            "DeviceID" : "CacheMemory:1",
            "ElementName" : "L1 - Cache",
            "EnabledState" : "Enabled",
            "HealthState" : "OK",
            "InstanceNumber" : 1,
            "IsVolatileMemory" : "Yes",
            "Name" : "Memory",
            "RequestedState" : "Not Applicable",
            "TotalMemory" : "393216 Bytes"
        },
        {
            "@odata.id" : "/DASH/v1/MemoryCollection/Memory",
            "@odata.type" : "#Memory.v1_0_0.Memory",
            "AccessType" : "Read/Write Supported",
            "AvailableMemory" : "2048 KB",
            "DeviceID" : "CacheMemory:2",
            "ElementName" : "L2 - Cache",
            "EnabledState" : "Enabled",
            "HealthState" : "OK",
            "InstanceNumber" : 2,
            "IsVolatileMemory" : "Yes",
            "Name" : "Memory",
            "RequestedState" : "Not Applicable",
            "TotalMemory" : "2048 KB"
        },
        {
            "@odata.id" : "/DASH/v1/MemoryCollection/Memory",
            "@odata.type" : "#Memory.v1_0_0.Memory",
            "AccessType" : "Read/Write Supported",
            "AvailableMemory" : "4096 KB",
            "DeviceID" : "CacheMemory:3",
            "ElementName" : "L3 - Cache",
            "EnabledState" : "Enabled",
            "HealthState" : "OK",
            "InstanceNumber" : 3,
            "IsVolatileMemory" : "Yes",
            "Name" : "Memory",
            "RequestedState" : "Not Applicable",
            "TotalMemory" : "4096 KB"
        }
    ],
    "Name" : "Memory Collection"
}
</pre>

</td></tr>
<tr><td>Output Class</td><td>

```
public class MemoryCollection {
    [JsonProperty("@odata.id")]
    public string _odata_id { get; set; }
    [JsonProperty("@odata.type")]
    public string _odata_type { get; set; }
    public string Id { get; set; }
    [JsonProperty("Members")]
    public List<MemoryMember> Members { get; set; }
    public string Name { get; set; }
```

</td></tr>
</table>

	}
	public class MemoryMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string AccessType { get; set; }
	public string AvailableMemory { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string IsVolatileMemory { get; set; }
	public string Name { get; set; }
	public string RequestedState { get; set; }
	public string TotalMemory { get; set; }
	}

37.5 Enumerate Sensor

Lists sensors associated with a specific processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0] enumerate sensor`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]", "Commands":["enumerate","sensor"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/SensorCollection",
	"@odata.type" : "#SensorCollection.v1_0_0.SensorCollection",
	"Id" : "SensorCollection",
	"Members" : [
	{
	"@odata.id" : "/DASH/v1/SensorCollection/Sensor",
	"@odata.type" : "#Sensor.v1_0_0.Sensor",
	"BaseUnit" : "Degrees C",
	"CurrentState" : "Normal",
	"DeviceID" : "Sensor:0",
	"ElementName" : "APU_Temperature",
	"EnabledState" : "Enabled",
	"HealthState" : "OK",
	"InstanceNumber" : 0,
	"Name" : "Sensor",
	"OperationalState" : ["OK"],
	"PossibleStates" : ["Normal", "Unknown"],
	"RateUnits" : "None",
	"RequestedState" : "No Change",
	"SensorReading" : 43,
	"SensorType" : "Temperature",
	"ThresholdLC" : "N/A",
	"ThresholdLF" : "N/A",
	"ThresholdLNC" : "N/A",
	"ThresholdUC" : "N/A",
	"ThresholdUF" : "N/A",
	"ThresholdUNC" : "N/A",
	"UnitModifier" : 0
	}
	],
	"Name" : "Sensor Collection"
	}

Output Class	public class SensorCollection { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string _odata_type { get; set; } public string Id { get; set; } [JsonProperty("Members")]

	public List<SensorMember> Members { get; set; }
	public string Name { get; set; }
	}
	public class SensorMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public string BaseUnit { get; set; }
	public string CurrentState { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public List<string> PossibleStates { get; set; }
	public string RateUnits { get; set; }
	public string RequestedState { get; set; }
	public int SensorReading { get; set; }
	public string SensorType { get; set; }
	public int ThresholdLC { get; set; }
	public int ThresholdLF { get; set; }
	public int ThresholdLNC { get; set; }
	public int ThresholdUC { get; set; }
	public int ThresholdUF { get; set; }
	public int ThresholdUNC { get; set; }
	public int UnitModifier { get; set; }
	}

37.6 Asset Show

Shows a specific asset linked to a processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0]/asset[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/asset[0]", "Commands":["show"]}`

Output JSON	`{`
	`"@odata.id" : "/DASH/v1/PhysicalAssetCollection/PhysicalAsset",`
	`"@odata.type" : "#PhysicalAsset.v1_0_0.PhysicalAsset",`
	`"CanbeFRUed" : "No",`
	`"ElementName" : "Processor #1",`
	`"InstanceNumber" : 0,`
	`"Manufacturer" : "AMD",`
	`"Model" : "AMD Phenom(tm) II X4 B95 Processor",`
	`"Name" : "PhysicalAsset",`
	`"PartNumber" : "N/A",`
	`"SKU" : "N/A",`
	`"SerialNumber" : "N/A",`
	`"Tag" : "BRCM:4.1"`
	`}`

Output Class	public class PhysicalAssetMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string ConnectorLayout { get; set; } public List<string> ConnectorType { get; set; } public string ElementName { get; set; } public int InstanceNumber { get; set; } public string Manufacturer { get; set; } public string Model { get; set; } public string Name { get; set; } public string PartNumber { get; set; } public string SKU { get; set; }

	public string SerialNumber { get; set; } public int SlotNumber { get; set; } public string Tag { get; set; } public string CanbeFRUed { get; set; } public string MemoryBankLabel { get; set; } public string MemoryCapacity { get; set; } public string MemoryFormFactor { get; set; } public string MemoryType { get; set; } public string IsHostingBoard { get; set; } public string PackageType { get; set; } public string ChassisType { get; set; } }

37.7 Fan

Shows a specific fan linked to a processor.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t processor[0]/fan[0] show
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/fan[0]", "Commands":["show"]}
Output JSON	{ "@odata.id" : "/DASH/v1/FanCollection/Fan", "@odata.type" : "#Fan.v1_0_0.Fan", "DesiredSpeed" : "N/A", "DeviceID" : "BRCM:10.1", "ElementName" : "Fan #1", "EnabledState" : "Not Applicable", "HealthState" : "OK", "InstanceNumber" : 0, "IsActiveCooling" : "Yes", "IsVariableSpeed" : "Yes", "Name" : "Fan", "OperationalState" : ["OK"], "RequestedState" : "Not Applicable" }

Output Class	public class FanMember {
	[JsonProperty("@odata.id")]
	public string _odata_id { get; set; }
	[JsonProperty("@odata.type")]
	public string odatatype { get; set; }
	public int DesiredSpeed { get; set; }
	public string DeviceID { get; set; }
	public string ElementName { get; set; }
	public string EnabledState { get; set; }
	public string HealthState { get; set; }
	public int InstanceNumber { get; set; }
	public string IsActiveCooling { get; set; }
	public string IsVariableSpeed { get; set; }
	public string Name { get; set; }
	public List<string> OperationalState { get; set; }
	public string RequestedState { get; set; }
	}

Note: List of Fan commands

Command	Input JSON command
Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/fan[0]", "Commands":["show"]}
Enumerate Asset	{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/fan[0]", "Commands":["enumerate","asset"]}
Asset Show	{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/fan[0]/asset[0]", "Commands":["show"]}

37.8 Memory

Shows a specific memory device linked to a processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0]/memory[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/memory[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/MemoryCollection/Memory", "@odata.type" : "#Memory.v1_0_0.Memory", "AccessType" : "Read/Write Supported", "AvailableMemory" : "4096 MB", "DeviceID" : "BRCM:8.1", "ElementName" : "Total System Memory", "EnabledState" : "Enabled", "HealthState" : "Unknown", "InstanceNumber" : 0, "IsVolatileMemory" : "Yes", "Name" : "Memory", "RequestedState" : "Not Applicable", "TotalMemory" : "4096 MB" }`
Output Class	`public class MemoryMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string AccessType { get; set; } public string AvailableMemory { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string IsVolatileMemory { get; set; } public string Name { get; set; } public string RequestedState { get; set; } public string TotalMemory { get; set; } }`

Note: List of Memory commands

Command	Input JSON command
Show	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/memory[0]", "Commands":["show"]}`
Enumerate Asset	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/memory[0]", "Commands":["enumerate","asset"]}`
Asset Show	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/memory[0]/asset[0]", "Commands":["show"]}`

37.9 Sensor Show

Shows a specific sensor linked to a processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t processor[0]/sensor[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/sensor[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/SensorCollection/Sensor", "@odata.type" : "#Sensor.v1_0_0.Sensor", "BaseUnit" : "RPM", "CurrentState" : "Normal", "DeviceID" : "BRCM:11.1", "ElementName" : "Numeric Sensor #1 (Tachometer)", "EnabledState" : "Not Applicable", "HealthState" : "Unknown", "InstanceNumber" : 0,`

<table>
<tr><td></td><td><pre>
"Name" : "Sensor",
"OperationalState" : ["No Contact"],
"PossibleStates" : [
"Non-Critical",
"Lower Non-Critical",
"Upper Non-Critical",
"Critical",
"Lower Critical",
"Upper Critical",
"Fatal",
"Lower Fatal",
"Upper Fatal",
"Normal",
"Unknown"
],
"RateUnits" : "None",
"RequestedState" : "No Change",
"SensorReading" : 0,
"SensorType" : "Tachometer",
"ThresholdLC" : "N/A",
"ThresholdLF" : "N/A",
"ThresholdLNC" : "N/A",
"ThresholdUC" : "N/A",
"ThresholdUF" : "N/A",
"ThresholdUNC" : "N/A",
"UnitModifier" : 0
}
</pre></td></tr>
<tr><td>Output Class</td><td><pre>
public class SensorMember {
[JsonProperty("@odata.id")]
public string _odata_id { get; set; }
[JsonProperty("@odata.type")]
public string odatatype { get; set; }
public string BaseUnit { get; set; }
public string CurrentState { get; set; }
public string DeviceID { get; set; }
public string ElementName { get; set; }
public string EnabledState { get; set; }
public string HealthState { get; set; }
public int InstanceNumber { get; set; }
public string Name { get; set; }
public List<string> OperationalState { get; set; }
public List<string> PossibleStates { get; set; }
public string RateUnits { get; set; }
public string RequestedState { get; set; }
public int SensorReading { get; set; }
public string SensorType { get; set; }
public int ThresholdLC { get; set; }
public int ThresholdLF { get; set; }
public int ThresholdLNC { get; set; }
public int ThresholdUC { get; set; }
public int ThresholdUF { get; set; }
public int ThresholdUNC { get; set; }
public int UnitModifier { get; set; }
}
</pre></td></tr>
</table>

Note: List of Sensor commands

Command	Input JSON command
Show	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/sensor[0]", "Commands":["show"]}`
Reading	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/sensor[0]", "Commands":["reading"]}`
Set Threshold	`{"h":"dash-system","u":"admin","P":"adminpass","t":"processor[0]/sensor[0]", "Commands":["setthreshold","lnc","100"]}`

38 Record Log

Purpose: Recorded log collection management.

38.1 Record Log Show

Shows details for a specific recorded log.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t recordlog[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"recordlog[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/RecordLogCollection/RecordLog", "@odata.type" : "#RecordLog.v1_0_0.RecordLog", "CurrentNumberOfRecords" : 75, "ElementName" : "Event Log", "EnabledState" : "Enabled", "HealthState" : "N/A", "InstanceID" : "BRCM:70.1", "InstanceNumber" : 0, "LogState" : "Normal", "MaxNumberOfRecords" : 499, "Name" : "Event Log", "OperationalStatus" : [0], "OverwritePolicy" : "Wraps When Full", "RequestedState" : "Not Applicable" }`
Output Class	`public class RecordLogMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public int CurrentNumberOfRecords { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string LogState { get; set; } public int MaxNumberOfRecords { get; set; } public string Name { get; set; } public List<int> OperationalStatus { get; set; } public string OverwritePolicy { get; set; } public string RequestedState { get; set; } }`

39 Registered Profile

Purpose: DMTF and other management profile registration.

39.1 Registered Profile Show

Shows details for a specific registered profile.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t registeredprofile[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"registeredprofile[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/RegisteredProfileCollection/RegisteredProfile", "@odata.type" : "#RegisteredProfile.v1_0_0.RegisteredProfile", "AdvertiseTypeDescriptions" : "N/A", "AdvertiseTypes" : ["Not Advertised"], "InstanceID" : "AMD:1", "InstanceNumber" : 0, "Name" : "RegisteredProfile", "OtherRegisteredOrganization" : "N/A", "RegisteredName" : "Base Desktop and Mobile", "RegisteredOrganization" : "DMTF", "RegisteredVersion" : "1.0.0" }`
Output Class	`public class RegisteredProfileMember { public string odataid { get; set; } public string odatatype { get; set; } public string AdvertiseTypeDescriptions { get; set; } public string[] AdvertiseTypes { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string OtherRegisteredOrganization { get; set; } public string RegisteredName { get; set; } public string RegisteredOrganization { get; set; } public string RegisteredVersion { get; set; } }`

40 Roles

Purpose: User role definition and management.

40.1 Role Show

Shows details for a specific role.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t role[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":" role[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/RoleCollection/Role",
	"@odata.type" : "#Role.v1_0_0.Role",
	"BusinessCategory" : "Manageability",
	"Caption" : "system:0",
	"CommonName" : "DASH:Administrator",
	"Description" : "system:0",
	"ElementName" : "Role:0",
	"InstanceID" : "Role:0",
	"InstanceNumber" : -1,
	"Name" : "Role:0",
	"RoleCharacteristics" :
	[
	"Opaque"
	],
	"displayPrivileges" : "Role:0"
	}

Output Class	public class RoleMember
	{
	public string odataid { get; set; }
	public string odatatype { get; set; }
	public string BusinessCategory { get; set; }
	public string Caption { get; set; }
	public string CommonName { get; set; }
	public string Description { get; set; }
	public string ElementName { get; set; }
	public string InstanceID { get; set; }
	public int InstanceNumber { get; set; }
	public string Name { get; set; }
	public string[] RoleCharacteristics { get; set; }
	public string displayPrivileges { get; set; }
	}

41 Sensor

Purpose: Hardware sensor monitoring and threshold management.

41.1 Sensor Show

Shows detailed information for a specific sensor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t sensor[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"sensor[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/SensorCollection/Sensor",
	"@odata.type" : "#Sensor.v1_0_0.Sensor",
	"BaseUnit" : "RPM",
	"CurrentState" : "Normal",
	"DeviceID" : "BRCM:11.1",
	"ElementName" : "Numeric Sensor #1 (Tachometer)",
	"EnabledState" : "Not Applicable",
	"HealthState" : "Unknown",
	"InstanceNumber" : 0,
	"Name" : "Sensor",
	"OperationalState" : ["No Contact"],
	"PossibleStates" : [
	"Non-Critical",
	"Lower Non-Critical",
	"Upper Non-Critical",
	"Critical",
	"Lower Critical",
	"Upper Critical",
	"Fatal",
	"Lower Fatal",
	"Upper Fatal",
	"Normal",
	"Unknown"
	],
	"RateUnits" : "None",
	"RequestedState" : "No Change",
	"SensorReading" : 0,
	"SensorType" : "Tachometer",
	"ThresholdLC" : "N/A",
	"ThresholdLF" : "N/A",
	"ThresholdLNC" : "N/A",
	"ThresholdUC" : "N/A",
	"ThresholdUF" : "N/A",
	"ThresholdUNC" : "N/A",
	"UnitModifier" : 0
	}

Output Class	public class SensorMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string BaseUnit { get; set; } public string CurrentState { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string HealthState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public List<string> OperationalState { get; set; } public List<string> PossibleStates { get; set; } public string RateUnits { get; set; } public string RequestedState { get; set; } public int SensorReading { get; set; }

	public string SensorType { get; set; } public int ThresholdLC { get; set; } public int ThresholdLF { get; set; } public int ThresholdLNC { get; set; } public int ThresholdUC { get; set; } public int ThresholdUF { get; set; } public int ThresholdUNC { get; set; } public int UnitModifier { get; set; } }

41.2 Sensor Reading

Retrieves the current reading from a specific sensor.

DASH CLI Command	**dashcli -h dash-system -u admin -P adminpass -t sensor[0] reading**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":"dash-system","u":"admin","P":"adminpass","t":"sensor[0]", "Commands":["reading"]}**
Output JSON	**{ "BaseUnit" : "RPM", "CurrentReading" : 0 }**

Output Class	public class SensorReading {
	public string **BaseUnit** { get; set; }
	public int **CurrentReading** { get; set; }
	}

41.3 Sensor Set Threshold

Sets a threshold value for a specific sensor.

DASH CLI Command	**dashcli -h dash-system -u admin -P adminpass -t sensor[0] setthreshold lnc 100**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":"dash-system","u":"admin","P":"adminpass","t":"sensor[0]", "Commands":["setthreshold","lnc","100"]}**
Output JSON	**{ "ErrorCode" : 0, "ErrorMessage" : "Threshold Set Successfully" }**

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

42 Service Processor

Purpose: Management controller (service processor) information.

42.1 Service Processor Show

Shows detailed information for a specific service processor.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t serviceprocessor[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"serviceprocessor[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/ServiceProcessorCollection/ServiceProcessor", "@odata.type" : "#ServiceProcessor.v1_0_0.ServiceProcessor", "ElementNameEditSupported" : "N/A", "EnabledState" : "N/A", "InstanceNumber" : 0, "MaxElementNameLen" : "N/A", "Name" : "ServiceProcessor", "RequestedState" : "N/A" }`
Output Class	`public class ServiceProcessorMember { public string odataid { get; set; } public string odatatype { get; set; } public string ElementNameEditSupported { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string MaxElementNameLen { get; set; } public string Name { get; set; } public string RequestedState { get; set; } }`

43 Software

Purpose: Software/firmware inventory and management.

43.1 Software Show

Shows detailed information for a specific software instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t software[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"software[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/SoftwareCollection/Software", "@odata.type" : "#Software.v1_0_0.Software", "BuildNumber" : "N/A", "ElementName" : "System Firmware Information", "IdentityInfoType" : "N/A", "IdentityInfoValue" : "N/A", "InstanceID" : "BRCM:29.1", "InstanceNumber" : 0, "IsEntity" : "True", "MajorVersion" : 1, "Manufacturer" : "Hewlett-Packard", "MinorVersion" : 11, "Name" : "Software", "OperationalStatus" : ["Unknown"], "RevisionNumber" : "N/A", "SoftwareClassifications" : ["Firmware", "Firmware/BIOS", "BIOS/FCode"], "TargetedOperatingSystems" : "N/A", "VersionString" : "786G6 v01.11" }`
Output Class	`public class SoftwareMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public object BuildNumber { get; set; } public string ElementName { get; set; } public string IdentityInfoType { get; set; } public string IdentityInfoValue { get; set; } public string InstanceID { get; set; } public int InstanceNumber { get; set; } public string IsEntity { get; set; } public int MajorVersion { get; set; } public string Manufacturer { get; set; } public int MinorVersion { get; set; } public string Name { get; set; } public List<string> OperationalStatus { get; set; } public object RevisionNumber { get; set; } public List<string> SoftwareClassifications { get; set; } public string TargetedOperatingSystems { get; set; } public string VersionString { get; set; } }`

43.2 Software Install

Initiates installation of software on the target system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t software[0] install http://10.138.141.209/xampp/bin/5762tm3.05`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"software[0]", "Commands":["install","http://10.138.141.209/xampp/bin/5762tm3.05"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "Software installation completed"`

	}

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

44 Software Update

Purpose: Software update process management.

44.1 Software Update Start

Starts a software update process on the target system.

DASH CLI Command	**dashcli -h dash-system -u admin -P adminpass softwareupdate start** **Choose valid instance number from the above list : 1** **Provide the File Path [http://{ip-address}[:{port}]/{path-to-firmware}/{firmware}] :http://10.138.141.209/xampp/bin/5762tm3.05**
jdo Usage	**dashcli -jdo**
URI	**https://<AMA_HOST>:<AMA_PORT>/**
Input JSON	**{"h":"dash-system","u":"admin","P":"adminpass", "Commands":["softwareupdate","start","1","http://10.138.141.209/xampp/bin/5762tm3.05"]}**
Output JSON	**{** **"ErrorCode" : 0,** **"ErrorMessage" : "Software installation completed"** **}**

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

45 SSH

Purpose: SSH service configuration management.

45.1 SSH Show

Shows detailed SSH configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t telnet[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"ssh[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/SSHCollection/SSH",
	"@odata.type" : "#SSH.v1_0_0.SSH",
	"AvailableRequestedStates" : "N/A",
	"Caption" : "SSH Protocol Endpoint Caption",
	"CommunicationStatus" : "Communication OK",
	"Description" : "SSH Protocol Endpoint Description",
	"DetailedStatus" : "Not Available",
	"ElementName" : "SSH Protocol Endpoint:0",
	"EnabledDefault" : "Enabled",
	"EnabledEncryptionAlgorithms" : "N/A",
	"EnabledSSHVersions" : "N/A",
	"EnabledState" : "Enabled",
	"Generation" : 77,
	"HealthState" : "OK",
	"IdleTimeout" : 300,
	"InstallDate" : "N/A",
	"InstanceID" : "SSH Protocol Endpoint:0",
	"InstanceNumber" : 0,
	"Name" : "SSH",
	"NameFormat" : "NameFormat",
	"OperatingStatus" : "Completed",
	"OperationalStatus" : "N/A",
	"OtherEnabledSSHVersion" : "OtherEnabledSSHVersion",
	"OtherEnabledState" : "None",
	"OtherEncryptionAlgorithm" : "N/A",
	"OtherSSHVersion" : "OtherSSHVersion",
	"OtherTypeDescription" : "IPv4",
	"PrimaryStatus" : "OK",
	"ProtocolIFType" : "IPv4",
	"ProtocolType" : "IPv4",
	"RequestedState" : "Enabled",
	"SSHVersion" : "SSHv2",
	"Status" : "OK",
	"StatusDescriptions" : "N/A",
	"SystemName" : "system:0",
	"TimeOfLastStateChange" : "N/A",
	"TransitioningToState" : "Enabled"
	}

Output Class	public class SSHMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string Name { get; set; } public string ElementName { get; set; } public string SystemName { get; set; } public string InstanceID { get; set; } public string Caption { get; set; } public string Description { get; set; } public string Status { get; set; } public string HealthState { get; set; } public string CommunicationStatus { get; set; } public string DetailedStatus { get; set; } public string OperatingStatus { get; set; }

```
        public string PrimaryStatus { get; set; }
        public string EnabledState { get; set; }
        public string OtherEnabledState { get; set; }
        public string RequestedState { get; set; }
        public string EnabledDefault { get; set; }
        public string TransitioningToState { get; set; }
        public string NameFormat { get; set; }
        public string ProtocolType { get; set; }
        public string ProtocolIFType { get; set; }
        public string OtherTypeDescription { get; set; }
        public string OtherEnabledSSHVersion { get; set; }
        public string OtherTerminalMode { get; set; }
        public string SSHVersion { get; set; }
        public string OtherSSHVersion { get; set; }
        public string InstallDate { get; set; }
        public string TimeOfLastStateChange { get; set; }
        public int Generation { get; set; }
        public int IdleTimeout { get; set; }
        public List<string> OperationalStatus { get; set; }
        public List<string> StatusDescriptions { get; set; }
        public List<string> AvailableRequestedStates { get; set; }
        public List<string> EnabledSSHVersions { get; set; }
        public List<string> EnabledEncryptionAlgorithms { get; set; }
        public List<string> OtherEncryptionAlgorithm { get; set; }
    }
```

46 Telnet

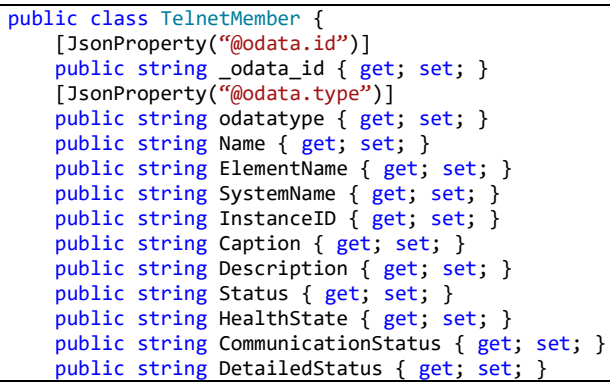
Purpose: Telnet service configuration management.

46.1 Telnet Show

Shows detailed Telnet configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t telnet[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"telnet[0]", "Commands":["show"]}`

Output JSON	{
	"@odata.id" : "/DASH/v1/TelnetCollection/Telnet",
	"@odata.type" : "#Telnet.v1_0_0.Telnet",
	"AvailableRequestedStates" : "N/A",
	"Caption" : "Telnet Protocol Endpoint Caption",
	"CommunicationStatus" : "Communication OK",
	"Description" : "Telnet Protocol Endpoint Description",
	"DetailedStatus" : "Not Available",
	"ElementName" : "Telnet Protocol Endpoint:0",
	"EnabledCharacterDisplayModes" : "N/A",
	"EnabledDefault" : "Enabled",
	"EnabledState" : "Enabled",
	"EnabledTerminalModes" : "N/A",
	"EscapeCharacter" : "-",
	"Generation" : 77,
	"HealthState" : "OK",
	"IdleTimeout" : 300,
	"InstallDate" : "N/A",
	"InstanceID" : "Telnet Protocol Endpoint:0",
	"InstanceNumber" : 0,
	"Name" : "Telnet",
	"NameFormat" : "NameFormat",
	"OperatingStatus" : "Completed",
	"OperationalStatus" : "N/A",
	"OtherCharacterDisplayMode" : "OtherCharacterDisplayMode",
	"OtherEnabledState" : "None",
	"OtherEnabledTerminalMode" : "N/A",
	"OtherTerminalMode" : "OtherTerminalMode",
	"OtherTypeDescription" : "IPv4",
	"PrimaryStatus" : "OK",
	"ProtocolIFType" : "IPv4",
	"ProtocolType" : "IPv4",
	"RequestedState" : "Enabled",
	"Status" : "OK",
	"StatusDescriptions" : "N/A",
	"SystemName" : "system:0",
	"TerminalMode" : "VT100",
	"TimeOfLastStateChange" : "N/A",
	"TransitioningToState" : "Enabled"
	}

Output Class	

```
        public string OperatingStatus { get; set; }
        public string PrimaryStatus { get; set; }
        public string EnabledState { get; set; }
        public string OtherEnabledState { get; set; }
        public string RequestedState { get; set; }
        public string EnabledDefault { get; set; }
        public string TransitioningToState { get; set; }
        public string NameFormat { get; set; }
        public string ProtocolType { get; set; }
        public string ProtocolIFType { get; set; }
        public string OtherTypeDescription { get; set; }
        public string TerminalMode { get; set; }
        public string OtherTerminalMode { get; set; }
        public string OtherCharacterDisplayMode { get; set; }
        public string EscapeCharacter { get; set; }
        public string InstallDate { get; set; }
        public string TimeOfLastStateChange { get; set; }
        public string BroadcastResetSupported { get; set; }
        public int Generation { get; set; }
        public int IdleTimeout { get; set; }
        public List<string> OperationalStatus { get; set; }
        public List<string> StatusDescriptions { get; set; }
        public List<string> AvailableRequestedStates { get; set; }
        public List<string> EnabledTerminalModes { get; set; }
        public List<string> OtherEnabledTerminalMode { get; set; }
        public List<string> EnabledCharacterDisplayModes { get; set; }
    }
```

47 Text Redirection

Purpose: Text console redirection management.

47.1 Text Redirection Show

Shows details for a specific text redirection instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t textredirection[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"textredirection[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/TextRedirectionCollection/TextRedirection", "@odata.type" : "#TextRedirection.v1_0_0.TextRedirection", "EnabledName" : "Text Redirection SAP for the Telnet Service", "EnabledState" : "Disabled", "InstanceNumber" : 0, "Name" : "TextRedirectionSAP:1", "PortNumber" : 87, "ProtocolType" : "Telnet", "RequestedState" : "Disabled", "TextFlowType" : "Terminal Mode" }`
Output Class	`public class TextRedirectionMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string EnabledName { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public int PortNumber { get; set; } public string ProtocolType { get; set; } public string RequestedState { get; set; } public string TextFlowType { get; set; } }`

47.2 Text Redirection Actions

Performs actions on a text redirection session (e.g., start/stop/activate/starttcr).

	<<option>>	
jdo Usage		`dashcli -jdo`
URI		`https://<AMA_HOST>:<AMA_PORT>/`
DASH CLI Command		`dashcli -h dash-system -u admin -P adminpass -t textredirection[0] <<option>>`
Input JSON		`{"h":"dash-system","u":"admin","P":"adminpass","t":"textredirection[0]", "Commands":["<<option>>"]}`
Output JSON	activate	`{ "ErrorCode" : 0, "ErrorMessage" : "Session activated successfully" }`
	disable	`{ "ErrorCode" : 0, "ErrorMessage" : "Session disabled successfully" }`
	start	`{ "ErrorCode" : 0, "ErrorMessage" : "Session Started successfully" }`
	starttcr	`{ "ErrorCode" : 0, "ErrorMessage" : "Session Started successfully" }`

DASH CLI Command		`dashcli -h dash-system -u admin -P adminpass textredirection <<option>>` `Choose valid instance number from the above list :0`
Input JSON		`{"h":"dash-system","u":"admin","P":"adminpass",` `"Commands":[" textredirection","<<option>>","0"]}`

Output JSON	connect	`{`
		`"ErrorCode" : 0,`
		`"ErrorMessage" : "Session Started successfully"`
		`}`

	disconnect	`{`
		`"ErrorCode" : 0,`
		`"ErrorMessage" : "Session disabled successfully"`
		`}`

Output Class		`public class CommandError {`
		`public int ErrorCode { get; set; }`
		`public string ErrorMessage { get; set; }`
		`}`

48 USB Redirection

Purpose: USB device redirection management.

48.1 USB Redirection Show

Shows details for a specific USB redirection instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t usbredirection[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"usbredirection[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/USBRedirectionCollection/USBRedirection", "@odata.type" : "#USBRedirection.v1_0_0.USBRedirection", "AccessURI" : "image", "ConnectionMode" : "Connect", "CreationClassName" : "CIM_USBRedirectionSAP", "ElementName" : "USB Redirection SAP", "EnabledState" : "Disabled", "InstanceNumber" : 0, "Name" : "USB Redirection", "RedirectionName" : "USBRedirectionSAP", "RequestedState" : "Disabled", "SystemCreationClassName" : "CIM_ComputerSystem", "SystemName" : "10.136.6.63" }`
Output Class	`public class USBRedirectionMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public string AccessURI { get; set; } public string ConnectionMode { get; set; } public string CreationClassName { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public string RedirectionName { get; set; } public string RequestedState { get; set; } public string SystemCreationClassName { get; set; } public string SystemName { get; set; } }`

48.2 USB Redirection Actions

Performs actions on a USB redirection session such as activate/disable/startvmr/startusbr/connect/disconnect.

	<<option>>	
jdo Usage		`dashcli -jdo`
URI		`https://<AMA_HOST>:<AMA_PORT>/`
DASH CLI Command		`dashcli -h dash-system -u admin -P adminpass -t usbredirection[0] <<option>>`
Input JSON		`{"h":"dash-system","u":"admin","P":"adminpass","t":"usbredirection[0]", "Commands":["<<option>>"]}`
Output JSON	activate	`{ "ErrorCode" : 0, "ErrorMessage" : "Session Activated successfully" }`

	disable	{
		"ErrorCode" : 0,
		"ErrorMessage" : "USB Redirection Service is Disabled Successfully"
		}

DASH CLI Command	startvmr	dashcli -h dash-system -u admin -P adminpass -t usbredirection[0] startvmr http://10.138.141.209/xampp/iso/dos6.iso false
Input JSON		{"h":"dash-system","u":"admin","P":"adminpass","t":"usbredirection[0]", "Commands":["startvmr","http://10.138.141.209/xampp/iso/dos6.iso","false"]}

Output JSON		{
		"ErrorCode" : 0,
		"ErrorMessage" : "USB Redirection Configuration has been successfully completed."
		}

DASH CLI Command	startusbr	dashcli -h dash-system -u admin -P adminpass -t usbredirection[0] startusbr \\\\.\\CdRom0 Note: supported only on Windows Management Console
Input JSON		{"h":"dash-system","u":"admin","P":"adminpass","t":"usbredirection[0]", "Commands":["startusbr","\\\\.\\CdRom0"]} Note: supported only on Windows Management Console

Output JSON		{
		"ErrorCode" : 0,
		"ErrorMessage" : "USB Redirection successful."
		}

DASH CLI Command	startbiosusbr	dashcli -h dash-system -u admin -P adminpass -t usbredirection[0] startbiosusbr \\\\.\\CdRom0 Note: supported only on Windows Management Console
Input JSON		{"h":"dash-system","u":"admin","P":"adminpass","t":"usbredirection[0]", "Commands":["startbiosusbr","\\\\.\\CdRom0"]} Note: supported only on Windows Management Console

Output JSON		{
		"ErrorCode" : 0,
		"ErrorMessage" : "USB Redirection successful."
		}

DASH CLI Command	connect	dashcli -h dash-system -u admin -P adminpass usbredirection connect Choose valid instance number from the above list :0 Using Integrate Web Server(Y/N) :n Provide the File Path [http://{ip-address}[:{port}]/{path-to-image}/{image}.iso] :http://10.138.141.209/xampp/iso/dos6.iso
Input JSON		{"h":"dash-system","u":"admin","P":"adminpass", "Commands":["usbredirection","connect","0","n","http://10.138.141.209/xampp/iso/dos6.iso"]}

Output JSON		{
		"ErrorCode" : 0,
		"ErrorMessage" : "USB Redirection Configuration has been successfully completed."
		}

DASH CLI Command	disconnect	dashcli -h dash-system -u admin -P adminpass usbredirection disconnect Choose valid instance number from the above list :0
Input JSON		{"h":"dash-system","u":"admin","P":"adminpass", "Commands":["usbredirection","disconnect","0"]}

Output JSON		{
		"ErrorCode" : 0,
		"ErrorMessage" : "USB Redirection Service is Disabled Successfully"
		}

Output Class		public class CommandError {
		public int ErrorCode { get; set; }
		public string ErrorMessage { get; set; }
		}

Note: As a special case, startusbr and startbiosusbr additionally support stream-based response as these commands take a long time to complete the workflow.

Since streaming response consists of multiple JSON responses sent over time, this request will need to be strictly implemented as a Post request.

The request will have changes in both JSON as well as connection processing.

The JSON will need to include a flag “serial” with value 1.

JSON Input	{“h”:”dash-system”,”u”:”admin”,”P”:”adminpass”,”t”:”usbredirection[0]”, ”stream”:1, “Commands”:[“startusbr”,”\\\\.\\CdRom0”]}

JSON Output	data: {"correlationId":"8a44282942d8e685-0","host":"dash-system","message":"[1/2] Enabling USBR SAP","phase":"enable-sap","severity":"info","stepIndex":1,"stepTotal":2,"terminal":false,"timestamp":1784616914433}
	
	data: {"correlationId":"8a44282942d8e685-0","host":"dash-system","message":"[2/2] Running client application","phase":"client-app","severity":"info","stepIndex":2,"stepTotal":2,"terminal":false,"timestamp":1784616915366}
	
	{
	“ErrorCode” : 0,
	“ErrorMessage” : “USB Redirection successful.”
	}

The JSON streamed response will have a correlationId tag that ensures that the response sent to the destination for a given request maintains the same Id throughout via a mechanism called Server-Sent Events (SSE).

A console that requires streaming support needs to make the corresponding changes to process streamed data till a response is obtained.

If user is using curl to send AMA request, then the Curl command will need few additional flags to prevent buffering, specify Post and allow streaming.

Without SSE	echo {“h”:”dash-system”,”u”:”admin”,”P”:”adminpass”,”t”:”usbredirection[0]”, “Commands”:[“startusbr”,”\\\\.\\CdRom0”]} | curl -k -N -H "Accept: text/event-stream" -X POST -d @- https://127.0.0.1:8080
With SSE	echo {“h”:”dash-system”,”u”:”admin”,”P”:”adminpass”,”t”:”usbredirection[0]”, ”stream”:1, “Commands”:[“startusbr”,”\\\\.\\CdRom0”]} | curl -k -N -H "Accept: text/event-stream" -X POST -d @- https://127.0.0.1:8080

49 User

Purpose: User account management

49.1 User Show

Shows details for a specific DASH user account.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t user[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"user[0]", "Commands":["show"]}`
Output JSON	`{ "@odata.id" : "/DASH/v1/ManagerAccountCollection/ManagerAccount", "@odata.type" : "#ManagerAccount.v1_0_0.ManagerAccount", "AssociatedRoles" : ["Admin", "Redirection", "Basic"], "ElementName" : "Account", "EnabledState" : "Not Applicable", "InstanceNumber" : 0, "Name" : "Administrator", "OrganizationName" : ["DMTF"], "RequestedState" : "Not Applicable", "UserId" : "Administrator" }`
Output Class	`public class UserMember { [JsonProperty("@odata.id")] public string _odata_id { get; set; } [JsonProperty("@odata.type")] public string odatatype { get; set; } public List<string> AssociatedRoles { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public int InstanceNumber { get; set; } public string Name { get; set; } public List<string> OrganizationName { get; set; } public string RequestedState { get; set; } public string UserId { get; set; } }`

49.2 User Add

Adds a new DASH user account to the system.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t computersystem[0] user add userName userPassword organizationName`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"computersystem[0]", "Commands":["user","add","userName","userPassword","organizationName"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "User Added Successfully" }`
Output Class	`public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

49.3 User Enable

Enables a specific DASH user account.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t user[2] enable`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"user[2]", "Commands":["enable"]}`

Output JSON	{
	"ErrorCode" : 0,
	"ErrorMessage" : "User Enabled Successfully."
	}

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

49.4 User Disable

Disables a specific DASH user account.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t user[2] disable
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"user[2]", "Commands":["disable"]}

Output JSON	{
	"ErrorCode" : 0,
	"ErrorMessage" : "User Disabled Successfully."
	}

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

49.5 User Assign Roles

Assigns roles to a specific DASH user.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t user[2] assignroles Role:0 Role:1
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"user[2]", "Commands":["assignroles","Role:0","Role:1"]}

Output JSON	{
	"ErrorCode" : 0,
	"ErrorMessage" : "Roles assigned Successfully."
	}

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

49.6 User Remove Roles

Removes roles from a specific DASH user.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t user[2] removeroles Role:1
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"user[2]", "Commands":["removeroles","Role:1"]}

Output JSON	{
	"ErrorCode" : 0,
	"ErrorMessage" : "Roles removed Successfully."
	}

Output Class	public class CommandError {
	public int ErrorCode { get; set; }
	public string ErrorMessage { get; set; }
	}

49.7 User Delete

Deletes a specific DASH user account from the system.

DASH CLI Command	dashcli -h dash-system -u admin -P adminpass -t user[2] delete
jdo Usage	dashcli -jdo

URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","t":"user[2]", "Commands":["delete"]}`
Output JSON	`{ "ErrorCode" : 0, "ErrorMessage" : "User Deleted Successfully." }`
Output Class	`public class CommandError { public int ErrorCode { get; set; } public string ErrorMessage { get; set; } }`

50 Wi-Fi Port

Purpose: Wi-Fi network port configuration and status.

50.1 Wi-Fi Port Show

Shows detailed Wi-Fi port configuration for a specific instance.

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -t wifiport[0] show`
jdo Usage	`dashcli -jdo`
URI	`https://<AMA_HOST>:<AMA_PORT>/`
Input JSON	`{"h":"dash-system", "u":"admin","P":"adminpass","t":"wifiport[0]","Commands":["show"]}`
Output JSON	`{ "@odata.id": "/DASH/v1/WiFiPortCollection", "@odata.type": "#WiFiPortCollection.v1_0_0.WiFiPortCollection", "Id": "WiFiPortCollection", "Members": [{ "@odata.id": "/DASH/v1/WiFiCollection/WiFiPort", "@odata.type": "#WiFiPort.v1_0_0.WiFiPort", "AuthenticationMethod": "WPA2 PSK", "ChannelNumber": 36, "DeviceID": "21", "ElementName": "WiFiPort:0", "EnabledState": "Enabled", "EncryptionMethod": "CCMP", "FrequencyBand": "5 GHz", "InstanceNumber": 0, "LinkTechnology": "Wireless LAN", "MACAddress": "00:03:7f:29:67:61", "MaximumSpeed": "N/A", "Name": "Qualcomm WCN685x Wi-Fi 6E Dual Band Simultaneous (DBS) WiFiCx Network Adapter #9", "NetworkType": "Infrastructure", "PortType": "802.11ax (Wi-Fi 6E)", "RequestedSpeed": "N/A", "RequestedState": "Not Applicable", "SSID": "Aruba_WPA2_PSK_5G", "SignalStrength": "88%", "Speed": "1 Gbps" }], "Name": "WiFiPort Collection" }`
Output Class	`public class WiFiPortMember { public string odataid { get; set; } public string odatatype { get; set; } public string AuthenticationMethod { get; set; } public int ChannelNumber { get; set; } public string DeviceID { get; set; } public string ElementName { get; set; } public string EnabledState { get; set; } public string EncryptionMethod { get; set; } public string FrequencyBand { get; set; } public int InstanceNumber { get; set; } public string LinkTechnology { get; set; } public string MACAddress { get; set; } public string MaximumSpeed { get; set; } public string Name { get; set; } public string NetworkType { get; set; } public string PortType { get; set; } public string RequestedSpeed { get; set; } public string RequestedState { get; set; } public string SSID { get; set; } public string SignalStrength { get; set; } public string Speed { get; set; }`

	}

51 Parse XML

Purpose: To decode and return the alert in SOAP XML format as Key Value JSON data to populate the output of alert messages on the management console.

51.1 Parse Alert

Parse alert and return back the Alert Message Info.

DASH CLI Command	N/A
jdo Usage	dashcli -jdo
URI	https://<AMA_HOST>:<AMA_PORT>/
Input JSON	{"h":"dash-system","u":"admin","P":"adminpass","t":"wifiport[0]", "Commands":["show"]}
SOAP XML	<s:Envelope xmlns:s=\"http://www.w3.org/2003/05/soap-envelope\" xmlns:wsa=\"http://schemas.xmlsoap.org/ws/2004/08/addressing\" xmlns:wsman=\"http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd\" xmlns:cim=\"http://schemas.dmtf.org/wbem/wscim/common.xsd\"> <s:Header> <wsa:To>http://10.136.6.55:8080/eventsink</wsa:To> <wsa:Action s:MustUnderstand=\"true\">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AlertIndication</wsa:Action> </s:Header> <s:Body xmlns:i=\"http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AlertIndication\"> <i:CIM_AlertIndication> <i:AlertingManagedElement> <wsa:Address>/wsman</wsa:Address> <wsa:ReferenceParameters> <wsman:ResourceURI>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem</wsman:ResourceURI> <wsman:SelectorSet> <wsman:Selector Name=\"__cimnamespace\">BRCM/implementation </wsman:Selector> <wsman:SelectorName= \"CreationClassName\">CIM_ComputerSystem<wsman:Selector><wsman:Selector Name=\"Name\">10.138.143.54</wsman:Selector> </wsman:SelectorSet> </wsa:ReferenceParameters> </i:AlertingManagedElement> <i:AlertType>5</i:AlertType> <i:IndicationFilterName> BRCM:AllEventjhghgvs</i:IndicationFilterName> <i:IndicationIdentifier>BRCM:CIM:2be90e13-887b-03a3-155b-983bfdfd08af</i:IndicationIdentifier> <i:IndicationTime> <cim:CIM_DateTime>Thu Sep 14 15:08:18 2023</cim:CIM_DateTime> </i:IndicationTime> <i:MessageArguments>HP705</i:MessageArguments> <i:MessageID>PLAT0003</i:MessageID> <i:OwningEntity>DMTF</i:OwningEntity> <i:PerceivedSeverity>3</i:PerceivedSeverity> <i:ProbableCause>0</i:ProbableCause> <i:SystemCreationClassName>CIM_ComputerSystem</i:SystemCreationClassName> <i:SystemName>10.138.143.54</i:SystemName> </i:CIM_AlertIndication> </s:Body> </s:Envelope>
Input JSON	{"Commands":["parsealert"," <s:Envelope xmlns:s=\"http://www.w3.org/2003/05/soap-envelope\" xmlns:wsa=\"http://schemas.xmlsoap.org/ws/2004/08/addressing\" xmlns:wsman=\"http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd\" xmlns:cim=\"http://schemas.dmtf.org/wbem/wscim/common.xsd\"> <s:Header> <wsa:To>http://10.136.6.55:8080/eventsink</wsa:To> <wsa:Action s:MustUnderstand=\"true\">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AlertIndication</wsa:Action> </s:Header>

	<s:Body xmlns:i=\"http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AlertIndication\"> <i:CIM_AlertIndication> <i:AlertingManagedElement> <wsa:Address>/wsman</wsa:Address> <wsa:ReferenceParameters> <wsman:ResourceURI>http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem</wsman:ResourceURI> <wsman:SelectorSet> <wsman:Selector Name=\"__cimnamespace\">BRCM/implementation </wsman:Selector> <wsman:SelectorName= \"CreationClassName\">CIM_ComputerSystem<wsman:Selector><wsman:Selector Name=\"Name\">10.138.143.54</wsman:Selector> </wsman:SelectorSet> </wsa:ReferenceParameters> </i:AlertingManagedElement> <i:AlertType>5</i:AlertType> <i:IndicationFilterName> BRCM:AllEventjhghgvs</i:IndicationFilterName> <i:IndicationIdentifier>BRCM:CIM:2be90e13-887b-03a3-155b-983bfdfd08af</i:IndicationIdentifier> <i:IndicationTime> <cim:CIM_DateTime>Thu Sep 14 15:08:18 2023</cim:CIM_DateTime> </i:IndicationTime> <i:MessageArguments>HP705</i:MessageArguments> <i:MessageID>PLAT0003</i:MessageID> <i:OwningEntity>DMTF</i:OwningEntity> <i:PerceivedSeverity>3</i:PerceivedSeverity> <i:ProbableCause>0</i:ProbableCause> <i:SystemCreationClassName>CIM_ComputerSystem</i:SystemCreationClassName> <i:SystemName>10.138.143.54</i:SystemName> </i:CIM_AlertIndication> </s:Body> </s:Envelope>\"]}

Output JSON	{
	“AlertType”: “5”,
	“AlertingManagedElement”: “/wsmanhttp://schemas.dmtf.org/wbem/wscim/1/ cim-schema/2/CIM_ComputerSystemBRCM/ implementationCIM_ComputerSystem10.138.143.54”,
	“IndicationFilterName”: “ BRCM:AllEventjhghgvs”,
	“IndicationIdentifier”: “BRCM:CIM:2be90e13-887b-03a3-155b-983bfdfd08af”,
	“IndicationTime”: “Thu Sep 14 15:08:18 2023”,
	“MessageArguments”: “HP705”,
	“MessageID”: “PLAT0003”,
	“OwningEntity”: “DMTF”,
	“PerceivedSeverity”: “3”,
	“ProbableCause”: “0”,
	“SystemCreationClassName”: “CIM_ComputerSystem”,
	“SystemName”: “0.0.0.0”
	}

Output Class	public class ParseAlert
	{
	public string AlertType { get; set; }
	public string AlertingManagedElement { get; set; }
	public string IndicationFilterName { get; set; }
	public string IndicationIdentifier { get; set; }
	public string IndicationTime { get; set; }
	public string MessageArguments { get; set; }
	public string MessageID { get; set; }
	public string OwningEntity { get; set; }
	public string PerceivedSeverity { get; set; }
	public string ProbableCause { get; set; }
	public string SystemCreationClassName { get; set; }
	public string SystemName { get; set; }
	}

Appendix

A. HTTPS Sample commands

For Authentication scheme “S”, and port “p”, please surround the option with square brackets when running HTTPS commands. The default ports are:

Scheme	Port
HTTP	623
HTTPS	664

1) For ignoring self-signed certificates, please try the command with “C” as 1:

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -S https -p 664 -C enumerate computersystem`
jdo Usage	`dashcli -jdo`
URI	`/`
Input JSON	`{"h":"dash-system","u":"admin","P":"adminpass","S":["https"],"p":["664"],"C":1, "Commands":["enumerate","computersystem"]}`

2) For trusting self-signed certificates, please try the command with “C” as 0:

DASH CLI Command	`dashcli -h dash-system -u admin -P adminpass -S https -p 664 enumerate computersystem`
jdo Usage	`dashcli -jdo`
URI	`/`
Input JSON	`{"h":"dash-system ","u":"admin","P":"adminpass","S":["https"],"p":["664"],"C":0, "Commands":["enumerate","computersystem"]}`

B. Logging

Logging configuration and interpretation.
By default logs are written to the file:
C:\ProgramData\AMD\DASHService\AMDDASHService.log
While contacting support attaching this log will greatly benefit in debugging and root causing issues encountered in the application.

C. Error

Error code and Error Message mapping.

Error Code	Error Message
0	The operation completed successfully.
-1	The operation failed
-10	The target is not initialized
1	This operation is not supported by the DASH System.
2	An unknown error occurred. Use -v for details.
3	Cannot complete within timeout period.
4	Failed.
5	Invalid parameter.
6	In use.
4096	The operation was accepted and is being processed.
4097	The requested state change is not valid for the current state of the resource.
4098	Use of timeout parameter not supported.
4099	Busy.
4100	The command was not recognized.
4101	Number of parameters for the command are invalid
4102	The operation being requested was not performed because the user has not been authenticated.
4103	The domain user name or password is incorrect.
4104	Job queue thread is already running.
4105	Invalid Port specified by user
4106	Invalid credentials specified for Digest
4107	Invalid images assigned
4108	Invalid ACMS details specified by user
4109	Invalid instance number.
4110	No matching instance found.
4111	Invalid target path.
4112	Function not supported.
4113	Enumeration failed.
4114	Operation failed.
4115	No instances found on the remote system.
4116	Start IP address not specified.
4117	End IP address not specified.
4150	System Health instances are not found.
4200	The group was not found.
4201	The specified group does not exist.
4202	The operation cannot be performed on built-in group.
4203	The specified group already exists.

4204	The specified action is invalid, or incompatible with the selected group.
4300	The system cannot be removed from the default group.
4301	The specified system does not exist.
4302	The system already exists in the group.
4900	The configuration was not updated since one or more of the setting(s) is invalid.
4901	The configuration was not updated since one or more of the authentication parameters is invalid.
4902	The configuration was not updated since one or more of the transport parameters is invalid.
4903	The authentication account was not found.
5000	Invalid IP address.
5001	IP address is out of acceptable range.
5002	Could not establish network connection with the remote computer.
5003	Unauthorized. Verify authentication credentials or port configuration.
5004	Invalid Hostname
5005	Cannot resolve hostname. Verify the hostname or IP address is correct.
5006	Cannot connect to the remote system. The port may be closed or the system is unreachable.
5007	Connection timed out. A firewall may be blocking the connection.
5008	Cannot resolve proxy address. Verify proxy configuration.
5009	SOCKS5 proxy handshake failed.
5010	Unsupported protocol scheme. Use 'http' or 'https'.
5011	TLS handshake timed out. The system may be behind a firewall or not responding to TLS.
5012	TLS handshake failed. Verify TLS version compatibility and server certificate.
5013	Cannot load CA certificate file. Verify the file exists and is readable.
5014	Cannot load client certificate file. Verify the file exists and is in PEM format.
5015	Cannot load client key file. Verify the file exists and is in PEM format.
5016	Communication lost during request. The connection timed out or was dropped.
5017	No response received. The system may have stopped responding or the connection was dropped.
5018	Access denied. The account may be locked or disabled on the DASH System.
5019	The DASH System returned an internal error (HTTP 500). Retry later or contact the system administrator.
5100	Power task failed.
5101	Power task not allowed on localsystem.
5102	Could not retrieve power management capabilities from the DASH System.
5150	Computer system operation failed.
5151	Adding user failed.
5152	Adding user function not supported.
5153	Adding boot config failed.
5154	Opaque management data creation failed.
5155	Power state change failed.
5156	System reboot failed.

6000	Text Redirection failed.
6001	Text Redirection disconnect failed.
6002	Text Redirection session is already in progress.
6003	Text Redirection activate failed.
6004	Text Redirection start failed.
6200	KVM Redirection failed.
6201	KVM Redirection not supported.
6202	KVM Redirection enumeration failed.
6203	KVM Redirection Status query failed.
6204	KVM Redirection enablement failed.
6205	KVM Redirection reboot failed.
6206	KVM Redirection connect failed.
6207	KVM Redirection timed out.
6208	KVM Redirection request aborted.
6209	Unable to perform operation, Verify if KVM Redirection is supported by the managed system.
6210	KVM Redirection controller timed out.
6211	KVM Redirection disable failed.
6212	Enumeration failed for IP Interface
6213	SSH Key file not found. Ensure 'KVMSSHKey' exists.
6214	Windows Recovery KVM is not supported on the platform.
6300	Client certificate file not found or unreadable. Verify the path and file permissions.
6301	Client key file not found or unreadable. Verify the path and file permissions.
6302	Mutual TLS authentication failed. The DASH System rejected the client certificate.
6303	Certificate authentication configuration error. Verify CertAuthConfig.json exists and is valid.
6304	No valid DASH certificate found in Windows Certificate Store. Automatic enrollment also failed.
6305	DASH certificate has expired. Re-enrollment attempted.
6306	ADCS certificate enrollment failed. Verify ADCS server is reachable and your account has enrollment permissions.
6307	Certificate in Windows Certificate Store has no associated private key. Re-import or re-enroll the certificate.
6308	Certificate with specified serial number not found in Windows Certificate Store.
6309	DASH System rejected the client certificate during TLS handshake.
6310	Failed to export certificate from Windows Store to PEM format.
6311	Certificate-store authentication (-a cert) is only supported on Windows. Use -cert and -key for file-based mutual TLS on Linux.
6400	Media Redirection failed
6401	Media Redirection Enumeration Failed
6402	Media Redirection Enable Failed
6403	Media Redirection Disable Failed
6404	Media Redirection Enable Failed as instance not found
6500	USB Redirection failed.

6501	USB Redirection disconnect failed.
6502	USB Redirection connect failed.
6503	USB Redirection connect mode is not connect.
6504	USB Redirection instance not found
6505	Length of URI specified is beyond allowed limit
6506	USB Redirection activate failed.
6507	USB Redirection start failed.
6508	Only .iso and .img files are supported for USB redirection.
6509	Failed to initialize ACMS cloud routing for USB redirection.
6510	USB Redirection not supported.
6700	Cannot connect to cloud relay. Verify the cloud relay address is correct and reachable.
6701	ACMS certificate file not found. Verify the -rc flag or default consolecert.pem exists.
6702	Target system is not registered with the cloud relay. Register the system first.
7000	Boot config failed.
7001	Boot config enumeration failed.
7002	Boot config change boot order failed.
7003	Boot device description string retrieval failed.
7004	Boot config change order failed.
7005	Boot config change order input failed.
7006	Failed to fetch Boot Config Instance
7007	Boot Device specified does not exist
7008	Size of boot device specified is larger than available boot device count
7009	Set Default operation has failed
7010	Set Next operation has failed
7011	Set Next Single Use operation has failed
8000	Indications failed.
8001	Enumeration of filter collections failed.
8002	Enumeration of subscribed indications failed.
8003	Subscription of indication failed.
8004	Unsubscription of indication failed.
8005	Subscription quota limit is reached.
8006	Master subscription filter list is empty.
8007	Deletion of alert destination failed.
8008	Deletion of indication filter failed.
8009	Deletion of filter collection failed.
8010	Renewal of subscription failed.
8011	Invalid event mode.
8012	Invalid expiration timeout number.
8013	Invalid heartbeat interval number.
8014	Creating alert destination failed.
8015	Creating indication filter failed.
8016	Creating filter collection failed.

8017	Maximum number of sessions supported not found.
8100	Battery operation failed.
8101	Battery enable failed.
8102	Battery disable failed.
8103	Battery test failed.
8104	Battery recharge failed.
8200	Log entry failed.
8201	Log entry enumeration failed.
8300	Sensor operation failed.
8301	Set threshold failed.
8302	Threshold reset is not supported by this sensor.
8303	Sensor subscription failed.
8304	Invalid threshold value.
8400	BIOS operation failed.
8401	BIOS restore defaults failed.
8402	BIOS set attribute failed.
8500	Software operation failed.
8501	Software install failed.
8600	User operation failed.
8601	Change password failed.
8602	Enable user failed.
8603	Disable user failed.
8604	Delete user failed.
8605	Assign roles failed.
8606	Remove roles failed.
8700	Role operation failed.
8701	Add permission failed.
8702	Set permission failed.
8703	Remove permission failed.
8704	Modification to this role is not permitted.
8705	Qualifier missing. Must be comma-separated in the permission.
8706	Role does not have this qualifier. Provide a valid permission for this role.
8800	Opaque Management Data operation failed.
8801	Read opaque management data failed.
8802	Write opaque management data failed.
8803	Delete opaque management data failed.
8804	Import opaque management data failed.
8805	Export opaque management data failed.
8806	Lock opaque management data failed.
8807	Reassign owner of opaque management data failed.
8808	Assign access to opaque management data failed.
8900	Memory operation failed.

8901	Memory instance not found.
9000	Fan operation failed.
9001	Fan instance not found.
9100	Power supply operation failed.
9101	Power supply instance not found.
9200	Processor operation failed.
9201	Processor instance not found.
9300	Power scheduler operation failed.
9301	Failed to create power scheduler.
9302	Failed to delete power scheduler.
9303	Failed to update power scheduler.
9304	Power scheduler limit of 3 reached. Delete an existing schedule to create a new one.
9305	Power scheduler name length is invalid.
9306	No power scheduler instances found.
9307	Power operation not allowed for power scheduler.
9400	Disk info operation failed.
9401	The attempt to erase/wipe disk was unsuccessful.
9402	Invalid disk password.
9403	The user account does not have access to perform this task.
10000	Invalid command '{}'
10001	Invalid target path at '{}'
10002	Invalid request to enumerate '{}'
10003	Invalid instance number in '{}'
10004	Error getting instance {}
10005	Invalid instance '{}'
10006	{} not found!
10007	Unable to open output file {}, GetLastError={}
10008	Error: Host name not specified for command {}
10009	Error: Invalid port number {}
10010	Error: Invalid scheme {}
10011	Error: Invalid option {}
10012	Error: Invalid timeout (must be 1-3600 seconds): {}
10013	Error: Invalid verbose number {}
10014	KVM Redirection {} {}
10015	Successfully scheduled to erase/wipe Disk: {}
10016	DiskDrive:{}
10017	No matching instance found in '{}'
10018	Invalid {} '{}'
10019	Error : Filter instance {} not found
10020	{} redirection type is not supported for this USB redirection session. {} is supported only if connection mode type is 'Connect'
10021	{}(s) not found

10022	Error: {} needs to follow {}
10023	Power state '{}' was applied successfully
10024	Power '{}' failed
10025	Software installation - {}
10026	Alert: There is already an active "{}" connection, close that connection before retrying this operation.
10027	Maximum Number of Sessions Supported = {}
95100	Boot To Text Power state change failed.
95101	Boot To Text Power task not allowed on localsystem.
95102	Boot To Text Power Management enumeration failed.
96000	Boot To Text failed to perform Text Redirection
96001	Boot To Text Text Redirection Disable failed
96002	Boot To Text failed to perform Text Redirection connect operation
96500	Boot To Text USB Instances are not found.
96501	Boot To Text URL disconnect failed.
96502	Boot To Text URL connect failed.
96503	Boot To Text Get instance failed.
97000	Boot To Text Boot Order change failed.
97001	Boot To Text Boot instances are not found.
97002	Boot To Text Boot config change boot order failed.
97003	Boot To Text Boot device description string retrieval failed.
97004	Boot To Text Boot config change order failed.
97005	Boot To Text Boot config change order input failed.
97006	Boot To Text Failed to fetch Boot Config Instance
97007	Boot To Text Boot Device specified does not exist
97008	Boot To Text Size of boot device specified is larger than available boot device count

D. Troubleshooting

Common issues and resolution steps.

D.1. Connection Issues

Symptom	Error Code	Possible Cause	Resolution
Cannot connect to remote system	5002	Network connectivity issue, wrong hostname/IP, or DASH service not running	Verify the hostname/IP is correct, ensure the system is powered on and reachable via ping, and confirm DASH service is running on the target
DNS resolution failure	5005	Hostname cannot be resolved	Use the IP address directly, or verify DNS configuration and hostname spelling
Connection refused	5006	DASH port is closed or system is unreachable	Verify the correct port is specified (-p flag). Default HTTPS port is 664. Ensure no firewall is blocking the port
Connection timed out	5007	Firewall blocking the connection	Check firewall rules on both client and target. Ensure the DASH port (default 664) is allowed
Unauthorized	5003	Invalid username or password	Verify credentials with -u (username) and -P (password) flags. Check if the account is locked on the DASH system
Access denied	5018	Account locked or disabled	Contact the system administrator to unlock or enable the account on the DASH system
HTTP 500 from DASH system	5019	Internal error on the managed system	Retry the operation. If persistent, reboot the DASH system or contact the system administrator

D.2. TLS / Certificate Issues

Symptom	Error Code	Possible Cause	Resolution
TLS handshake failed	5012	TLS version mismatch or certificate issue	Verify the target supports TLS 1.2+. Try with -C flag to skip certificate verification for testing
TLS handshake timed out	5011	Firewall blocking TLS or system not responding	Check firewall rules. Ensure the target system's DASH service is running
Cannot load CA certificate	5013	CA certificate file path is wrong or file is unreadable	Verify the file path passed to -c flag exists and is in PEM format

Cannot load client certificate	5014	Client certificate file is missing or not in PEM format	Verify the file path passed to -cert flag exists and is a valid PEM file
Cannot load client key	5015	Client key file is missing or not in PEM format	Verify the file path passed to -key flag exists and is a valid PEM file
Mutual TLS rejected	6302	DASH system rejected the client certificate	Verify the client certificate is signed by a CA trusted by the DASH system
Certificate not found in store	6304	No valid DASH certificate in Windows Certificate Store	Enroll a new certificate through ADCS, or use file-based mutual TLS with -cert and -key flags
Certificate expired	6305	DASH certificate has expired	Re-enroll the certificate through your certificate authority
Certificate serial not found	6308	Wrong serial number specified with -serial flag	List certificates in the Windows Certificate Store and verify the serial number
Cert auth Windows only	6311	Using -a cert on Linux	On Linux, use -cert and -key flags for file-based mutual TLS instead of -a cert

D.3. Cloud Relay (ACMS) Issues

Symptom	Error Code	Possible Cause	Resolution
Cannot connect to cloud relay	6700	Cloud relay address is wrong or unreachable	Verify the address passed to -r flag. Ensure the cloud relay server is running and accessible
ACMS certificate missing	6701	Certificate file not found	Verify the path passed to -cert flag or ensure the default consolecert.pem exists in the certs directory
Target not registered	6702	Target system not registered with cloud relay	Register the target system with the cloud relay before attempting connection

D.4. Proxy Issues

Symptom	Error Code	Possible Cause	Resolution
Cannot resolve proxy	5008	Proxy address is wrong	Verify the proxy address passed to -x flag
SOCKS5 handshake failed	5009	SOCKS5 proxy configuration issue	Verify the proxy server is running and supports SOCKS5 protocol

D.5. Command and Argument Errors

Symptom	Error Code	Possible Cause	Resolution
Invalid command	10000	Unrecognized command name	See the list of supported commands in this guide
Invalid option	10011	Unrecognized CLI flag	See the list of supported flags in this guide
Host not specified	10008	Missing -h flag	Add -h <hostname> to specify the target system
Invalid port number	10009	Port is not a valid number (0-65535)	Specify a valid port number between 1 and 65535
Invalid timeout	10012	Timeout value out of range	Specify a timeout between 1 and 3600 seconds
Flag argument missing	10022	A flag that requires a value was used without one	Provide the required argument after the flag (e.g., -h hostname)
Invalid instance number	4109	Instance number does not exist	Run enumerate first to list available instances and their numbers
No matching instance	4110	Instance not found on the target	Verify the target path and instance number. Run enumerate to list instances

D.6. Profile-Specific Issues

Symptom	Error Code	Possible Cause	Resolution
Power operation failed	5100	Power state change not supported or failed	Verify the target supports the requested power state. Use enumerate powermanagement to check capabilities
KVM not supported	6201	Target does not support KVM redirection	Verify KVM support with enumerate kvmredirection
KVM instance not found	6209	KVM service not available	The managed system may not support KVM redirection
Boot device does not exist	7007	Invalid boot device specified	Run enumerate bootconfig to list available boot devices
Operation not supported	1	DASH system does not support this operation	The target system firmware may not implement this DASH profile
Subscription limit reached	8005	Maximum number of indication subscriptions	Unsubscribe existing indications before creating new ones

D.7. General Troubleshooting Tips

Tip	Details

Enable verbose mode	Add -v 1 to any command to see detailed diagnostic output, including WS-Management request/response details
Skip certificate verification	Use -C flag to bypass TLS certificate verification during initial testing (not recommended for production)
Check version	Run version command to verify the installed AMA version
Default port	DASH systems typically listen on port 664 (HTTPS). If using HTTP, specify -S http -p 623
Timeout adjustment	For slow networks or cloud relay connections, increase the timeout with -T 120 (120 seconds)

A. Support

Support resources and contact information.

Discussion forum link

Link: http://www.amd.com/DASH

DASH support email

Email: dashsupport@amd.com