

## Release Note for AIC JBOD 4U60 BMC

Feb. 29th, 2024

### Changelog

02/29/2024 (4206JBD000600)

1. Modify the upper critical / upper non-recoverable threshold of the sensor PSUx\_Temp to 60 / 65 for power supply R1CA2122A and R1CA2132A.

01/25/2024 (4206JBD000500)

2. Fix bug: Fix the "FW Rev" of the Titanium power supply R1CA2801C and R1CA2132A.
3. Add SNMP Support.

10/31/2023 (4206JBD000400)

4. Fix bug: The SATA HDD shows "Not supported" at the Secondary canister.
5. Fix bug: Console can't login after the first admin's password changed

10/27/2023 (4206JBD000300)

6. Add RTC support
7. Add Zone Count function
8. Add HDD power on/off function
9. Fix bug: Default IP to 192.168.11.11
10. Fix bug: Chassis identify behavior (solid on -> blinking)
11. Fix bug: CPLD FW version reading
12. Fix bug: BMC alarm led when event occurs
13. Fix bug: Fan speed reading abnormal
14. Fix bug: Power restore policy function
15. Fix bug: Sensor Sys\_Temp-Bd won't lit the front panel temperature alarm led
16. Fix some Redfish bug
  - (1) PSU name: from PSU\_Left/Right to PSU1/2
  - (2) For chassis reset: remove "GracefulShutdown" and change "ForceRestart" from reset the hub expander to doing power cycle.

9/08/2023 (4206JBD000200)

1. Add CPLD Update function at Web UI.
2. Add redfish function

09/08/2023 (4206JBD000100)

1. Initial version.

## **Known Issue**

## 1. IPMI Command

### 1.1. IPMI standard Command

Only the IPMI standard commands with AIC specific implementation are listed below.

#### 1.1.1. Get Device ID (NetFn: 0x6, Command: 0x01)

Only the fields highlighted in green are AIC specific.

The firmware version is <Firmware Revision 1>.<Firmware Revision 2>.<Byte 1 of Auxiliary Firmware Revision Information>.

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2	Device ID
	3	Device Revision
	4	Firmware Revision 1 (Major Firmware Revision)
	5	Firmware Revision 2 (Minor Firmware Revision)
	6	IPMI Version
	7	Additional Device Support
	8:10	Manufacturer ID, LS Byte first.
	11:12	Product ID, LS Byte first.
	13	Byte 1 of Auxiliary Firmware Revision Information
	14	Byte 2 of Auxiliary Firmware Revision Information
	15	Byte 3 of Auxiliary Firmware Revision Information
	16	Byte 4 of Auxiliary Firmware Revision Information

### 1.2. IPMI OEM Command

#### 1.2.1 Get HDD Status (NetFn: 0x3C, Command: 0x31)

	Byte	Data Field
Request Data	-	-
Response Data	1	Completion Code
	2:25	Hard disk drive status for each slot 0x01: Normal 0x02: Abnormal 0x03: Absence 0x04: Presence but Power-off 0x05: Bad HDD 0x06: Not Supported

### 1.2.2. HDD Control (NetFn: 0x3C, Command: 0x32)

Request Data	Byte	Data Field
	1	Slot number
	2	Action 0x00: Drive off 0x01: Drive on
Response Data	1	Completion Code

### 1.2.3 Get BMC Firmware Name (NetFn: 0x3A, Command: 0x33)

Request Data	Byte	Data Field
	-	-
Response Data	1	Completion Code
	2:14	BMC Platform Name

## 2. Firmware Update with “Yafuflash” through TCP/IP Network

### 2.1. BMC firmware update (full upgrade without Config Module preserved)

\$ Yafuflash -nw -ip <BMC IP Address> -u <Username> -p <Password> -force-boot <BMC Firmware Filename> -mse 1

## 3. Firmware Update with “Redfish” through TCP/IP Network

The following APIs provides series of steps to update BMC firmware (full upgrade without Config Module preserved).

### 3.1. Change Password

Redfish Password change is required for Redfish Update Service and most of other URIs which require authorization.

Steps to change Redfish Password:

1. Select **Authorization** Type as **Basic Auth** and select the https method **PATCH**

URI: *https://{BMC IP}/redfish/v1/AccountService/Accounts/1*

Username: *Administrator*

Password: *superuser*

**PATCH** https://192.168.21.58/redfish/v1/AccountService/Accounts/1 Send

Params **Authorization** Headers (10) Body Pre-request Script Tests Settings Cookies

Type **Basic Auth**

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Username **Administrator**

Password **superuser**

2. Add **If-None-Match** header and provide the value "".

**PATCH** https://192.168.21.58/redfish/v1/AccountService/Accounts/1 Send

Params **Authorization** **Headers (10)** Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

Key	Value	Description
<input checked="" type="checkbox"/> <b>If-None-Match</b>	<b>""</b>	
Key	Value	Description

3. Select **Body** type as **raw** data type **JSON** and provide the following request body.

```
{
  "Password": "12345678"
}
```

**PATCH** https://192.168.21.58/redfish/v1/AccountService/Accounts/1 Send

Params **Authorization** Headers (10) **Body** Pre-request Script Tests Settings Cookies

**none** **form-data** **x-www-form-urlencoded** **raw** **binary** **GraphQL** **JSON**

```
1 {
2   "Password": "12345678"
3 }
```

4. Hit the **Send** button and make sure the response code is **204 No Content**

**PATCH** https://192.168.21.58/redfish/v1/AccountService/Accounts/1 Send

Params **Authorization** Headers (10) **Body** Pre-request Script Tests Settings Cookies

**none** **form-data** **x-www-form-urlencoded** **raw** **binary** **GraphQL** **JSON**

```
1 {
2   "Password": "12345678"
3 }
```

**Body** Cookies Headers (3) Test Results

Status: **204 No Content** Time: 167 ms Size: 128 B Save as Example

Pretty Raw Preview Visualize Text

1

5. Use the changed password as credentials for basic authorization to get the response for the Redfish URIs that require authentication.

### 3.2. BMC Firmware Update

Redfish API provides multipart push to update the BMC firmware and .bin BIOS image format.

#### 3.2.1. Multipart Push FW Update

Request:

```
POST /redfish/v1/UpdateService/upload
Host: <bmc_ip>
Content-Length: <len_of_request>
Content-Type: multipart/form-data;
boundary=-----493918603359346570222237
-----493918603359346570222237
Content-Disposition: form-data; name= UpdateFile ; filename=
encrypted_rom.ima
Content-Type: application/octet-stream
<image_binary>
-----493918603359346570222237
Content-Disposition: form-data; name= UpdateParameters ; filename=
parameters.json
Content-Type: application/json
{
  Targets :[
    /redfish/v1/UpdateService/FirmwareInventory/BMC
  ]
}
-----493918603359346570222237
Content-Disposition: form-data; name= OemParameters ; filename=
oem_paramters.json
Content-Type: application/json
{
  ImageType : BMC ,
}
-----493918603359346570222237--
```

The parameters are mentioned in 3.2.2.

Using POSTMAN, please follow the steps as given below.

1. Create a JSON file **parameters.json** with content like below.

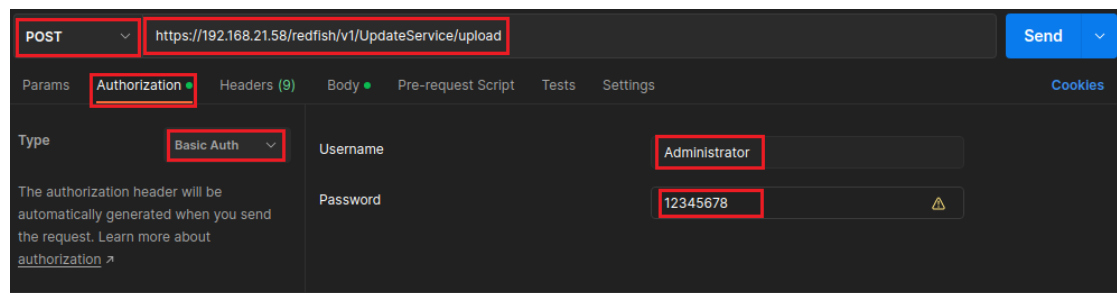
```
{
  "Targets":[
    "/redfish/v1/UpdateService/FirmwareInventory/BMCImage1"
  ]
}
```

2. Create a JSON file **oem\_parameters.json** with content like below.

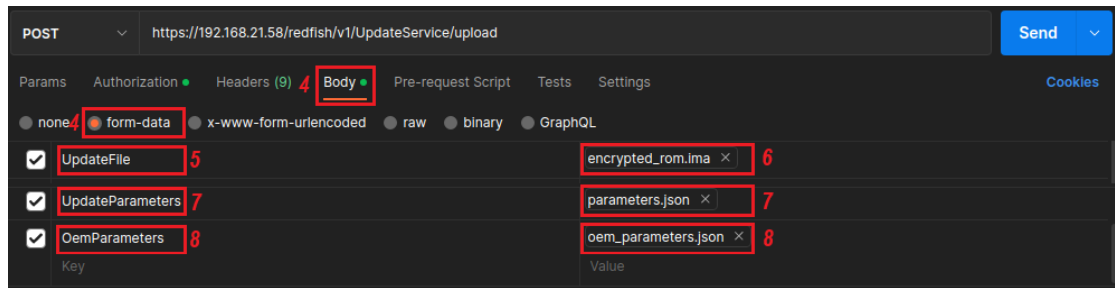
```
{
  "ImageType":"BMC"
}
```

3. Open POSTMAN and enter the URI to POST the request, select the https method **POST** and select **Authorization** Type as **Basic Auth**.

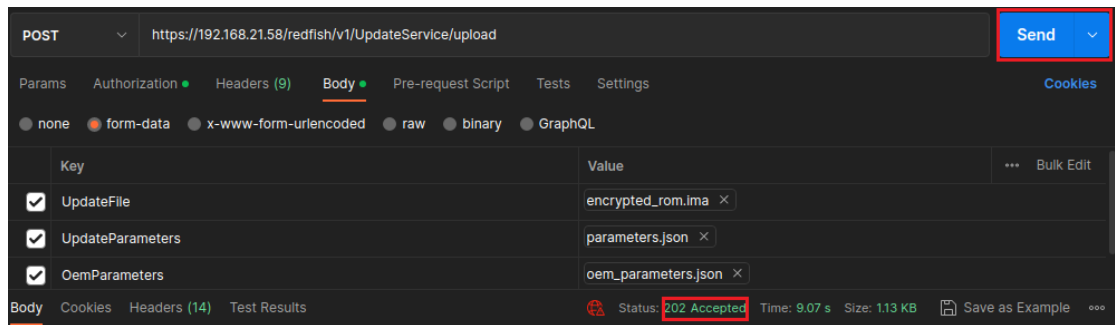
URI: *https://{BMC IP}/redfish/v1/UpdateService/upload*  
Username: *Administrator*  
Password: *12345678*



4. In the **Body** tab, select **form-data**.
5. Provide the key name as **UpdateFile** in the KEY section and change its type from Text to File.
6. Click on Select **File** in the VALUE section to select the BMC firmware image **.ima** file which is available in your local machine to update.
7. Like step 5 and 6, provide key name **UpdateParameters** and select **parameters.json** created in step 1.
8. Like step 5 and 6, provide key name **OemParameters** and select **oem\_parameters.json** created in step 2.

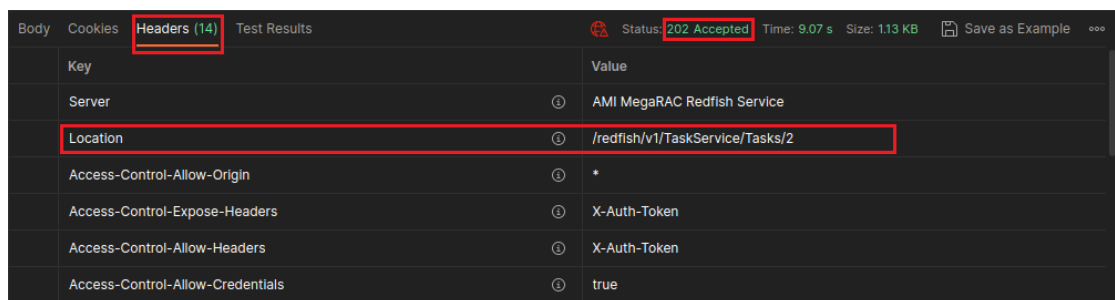


9. Hit the **Send** button and make sure the response code is **202 Accepted**



Response:

- Success: HTTP status will be **202 Accepted**. The server would create a Task and response status 202 after accepted this request, but the update has not been completed. The response http **Headers** would include a **Location** header. Location header has the link to the Task resource for this updating service.
- Failure: HTTP status will be other.



## 4. Redfish Resources

### 4.1 OEM Redfish Resources

#### 4.1.1 Expander Instance

Request:

```
GET https://{ip}/redfish/v1/Chassis/Self/Oem/Aic/Expander
```



Content-Type: application/json

Response:

Success: HTTP status will be **200 OK**. The response of the request will be in JSON format. The properties are mentioned in the following table.

Failure: HTTP status will be other.

Table-1 Expander Properties

Name	Type	ReadOnly	Description
@odata.context	String	True	Refer OData properties in Redfish API document.
@odata.id	String	True	Refer OData properties in Redfish API document.
@odata.type	String	True	Refer OData properties in Redfish API document.
@odata.etag	String	True	Refer OData properties in Redfish API document.
Id	String	True	Resource Identifier
Name	String	True	Name of the Resource
Description	String	True	Provides description of the resource.
Status	Object	True	This property represent if this resource is available or not and why.
Manufacturer	String	True	The vendor or manufacturer associated with this Expander.
Model	String	True	Model number of this Expander
Actions	String	True	The Actions object contains the available custom actions on this resource.
FirmwareRev	String	True	The firmware reversion of this Expander.
MFGRev	String	True	The configuration reversion of this Expander.
HDDStatus	Object	True	This object will contain the status of hard disk drive for each bay. Refer Table-2 HDDStatus properties.

Table-2 HDDStatus Properties

Name	Type	ReadOnly	Description	
Name	String	True	Name of the hard disk drive	
Status	String	True	Status of the hard disk drive.	
			Enum	Description
			Normal	This HDD is working normally.
			Abnormal	The HDD is working with an expected error.
			Absence	The HDD is not inserted or the device cannot be detected.
			Presence but Power-off	The HDD has been detected, but the slot is not powered.

			Bad HDD	An unexpected error occurred on this HDD.
--	--	--	---------	---