DELL EMC

White Paper

Updating Firmware using OpenManage Enterprise APIs

Abstract

This document shows the workflow and the APIs involved in the firmware update process for OpenManage Enterprise v3.0, v3.1, v3.2 and OpenManage Enterprise – Modular Edition v1.0 and v1.00.10

September 2019

Revisions

Date	Description
September 2019	Initial release

Acknowledgements

This paper was produced by the following:

Author: Vandana Mallempati

Support: OpenManage Enterprise and OpenManage Enterprise Modular engineering teams

Table of contents

Executive summary

This document shows the workflow and the APIs involved in the firmware update process for OpenManage Enterprise v3.0, v3.1, v3.2 and OpenManage Enterprise – Modular Edition v1.0 and v1.00.10.

1 Introduction

OpenManage Enterprise (OME) supports firmware update/downgrade actions across supported devices using one of the two possible workflows:

- Update/downgrade devices or a group (of devices) using a single DUP
- Update/downgrade devices or a group (of devices) using a catalog

2 Workflow

2.1 Update using a Catalog

1. Create a catalog from an online source or point to a remote repository that hosts a catalog and the associated DUP files referenced in the catalog. The online catalog source is https://downloads.dell.com. The supported offline catalog sources are:

   - HTTP
   - HTTPS
   - CIFS
   - NFS

2. Create a firmware baseline for the targets (device/devices/group of devices) using the catalog specified in step 1a.
3. Check the firmware baseline compliance report to see which components need to be updated / downgraded.
4. Update specific components and track the spawned job to completion.

2.2 Update using a single DUP

1. Upload the DUP file to OME and retrieve the file token returned on a successful upload.
2. Determine for the device or group if there are specific components that this DUP applies to for an update/downgrade.
3. Apply the DUP in the case of an update/downgrade to the specified components and parse the ID of the returned job that handles the application of the DUP file.
4. Track the job to completion and indicate errors / success.

2.3 Create firmware catalog

Catalogs are bundles of firmware based on device types.

Create the online catalog - All the available catalogs (update packages) are validated and posted to **support.dell.com**. When you create an online catalog, the catalog file is downloaded but the corresponding DUPs are not downloaded. When the devices are being updated, the DUP files are downloaded locally and deleted on successful completion of associated update tasks.

Customer-created catalogs (using the Dell Repository Manager tool for instance) can be stored on CIFS/NFS/HTTP/HTTPS shares.

3 APIs

POST /api/UpdateService/Catalogs

Description: This method creates a catalog

Result Codes:

HTTP Response Code	Description
201	Catalog successfully created

Privilege: Users with BASELINE_MANAGEMENT, JOB_MANAGEMENT and DEVICE_UPDATE privilege can create catalogs and baselines, and update firmware. These privileges map to the Admin user role.

Request:

Attribute Name	Type	Description	Notes
Filename	String	File name of the catalog	Not required for online catalog
SourcePath	String	Repository source (Full path including subfolders to where the catalog file is located)	Yes
Repository		**Repository details**	
Name	String	Name of the repository – The name must be unique for each repository that is created	Yes
Description	String	Description of the repository	
Source	String	URL or IP or FQDN of the repository host	Yes
DomainName	String	Domain Name for user credentials	
Username	String	Username to access the share containing the catalog (CIFS/HTTPS)	
Password	String	Password to access the share containing the catalog (CIFS/HTTPS)	
CheckCertificate	Boolean	If certificate check must be done for HTTPS repository	
RepositoryType	String	NFS/CIFS/HTTP/HTTPS/DELL_ONLINE	Yes

Example: Request Payload to create a catalog from Dell support site (downloads.dell.com)

```
{
 "Filename": "",
        "SourcePath": "",
        "Repository": {
            "Name": “Test”,
            "Description": "Dell Online test",
            "RepositoryType": "DELL_ONLINE",
            "Source": "downloads.dell.com",
            "DomainName": "",
            "Username": "",
            "Password": "",
            "CheckCertificate": False
        }
}
```

Example: Request Payload to create a catalog from a CIFS share

```
{
    "Filename":"Catalog.xml",
    "SourcePath":"DUPS\\R520\\",
    "Repository":{
                "Name":"CIFS",
                "Description":"CIFS Desc",
                "RepositoryType":"CIFS",
                "Source":”<ip address>",
                "DomainName":"americas",
                "Username":"administrator",
                "Password":"changeme",
                "CheckCertificate":false
    }
}
```

Example: Request Payload to create a catalog from an NFS share

```
{
    "Filename":"catalog.xml",
    "SourcePath":"/nfsshare/650/1150",
    "Repository":{
                "Name":"NFS-4",
                "Description":"NFS Desc",
                "RepositoryType":"NFS",
                "Source":"<ip address>",
                "DomainName":"",
                "Username":"",
                "Password":"",
                "CheckCertificate":false
    }
}
```

Example: Request Payload to create a catalog from a HTTPS location

```
{
    "Filename":"catalog.xml",
    "SourcePath":"install_packages/Packages",
    "Repository":{
                "Name":"HTTPS-again",
                "Description":"HTTPS Desc",
                "RepositoryType":"HTTPS",
                "Source":"<ip address:port_number>",
                "DomainName":"",
                "Username":"",
                "Password":"",
                "CheckCertificate":false
    }
}
```

Example: Response on successful catalog creation (HTTP 201)

Note: Catalog creation is an asynchronous process. Until the catalog is downloaded and all required assets are created, the returned ID values and catalog attributes are 0 or Null as appropriate. Users must determine the catalog ID by enumerating all catalogs and mapping to the one with the “Name” field used while creating the catalog.

```
{
    "Id": 0,
    "Filename": "",
    "SourcePath": "",
    "TaskId": 25108,
    "Status": null,
    "BaseLocation": null,
    "ManifestIdentifier": null,
    "ReleaseIdentifier": null,
    "ManifestVersion": null,
    "ReleaseDate": null,
    "LastUpdated": null,
    "BundlesCount": 0,
    "PredecessorIdentifier": null,
    "Repository": {
        "Id": 0,
        "Name": "Test",
        "Description": "",
        "Source": "downloads.dell.com",
        "DomainName": "",
        "Username": "",
        "Password": "",
        "CheckCertificate": false,
        "RepositoryType": "DELL_ONLINE"
    },
```

```
    "AssociatedBaselines": []
}
```

GET /api/UpdateService/Catalogs

Description: This method returns all existing catalogs

Result Codes:

HTTP Response Code	Description
200	Returns an enumeration of catalogs.

Privilege: View

Response:

Attribute Name	Type	Description
Filename	String	File name of the catalog
SourcePath	String	Relative path of the catalog file from the repository (for example: downloads.dell.com/catalog/catalog.gz)
Status	String	Status of the catalog creation
TaskId	Integer	The identifier of the task or job that is created to download the catalog
BaseLocation	String	The repository location of the catalog (for example, for an online catalog, the repository location is **downloads.dell.com**)
ManifestIdentifier	String	Catalog manifest identifier
ReleaseIdentifier	String	Catalog release identifier
ManifestVersion	String	Catalog manifest version
ReleaseDate	String	Catalog release date
LastUpdated	String	Date and time when the catalog instance was updated in the appliance
BundlesCount	Integer	Number of bundles referenced in the catalog
PredecessorIdentifier	String	Identifier of the catalog published before the current catalog
AssociatedBaselines		The baselines that are associated with the catalog. This is an array.
BaselineId	Integer	Identifier of the baseline associated with the catalog
BaselineName	String	Name of the baseline associated with the catalog
Repository details		**Description**

Attribute Name	Type	Description
Name	String	Name of the repository
Description	String	Description of the repository provided by users
Source	String	URL or location of the repository
DomainName	String	Domain for the user credentials (CIFS only)
Username	String	Username to access the share containing the catalog (CIFS/HTTPS)
Password	String	Password to access the share containing the catalog (CIFS/HTTPS)
CheckCertificate	String	Whether certificate checking was enabled for the HTTPS repository
RepositoryType	String	NFS/CIFS/HTTP/HTTPS/DELL_ONLINE

Example: Response Payload

Response payload showing a catalog created from Dell online. Baselines are not associated with this catalog yet.


```
{
    "@odata.context": "/api/$metadata#Collection(UpdateService.Catalogs)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#UpdateService.Catalogs",
            "@odata.id": "/api/UpdateService/Catalogs(22)",
            "Id": 22,
            "Filename": "catalog.xml",
            "SourcePath": "catalog/catalog.gz",
            "Status": "Completed",
            "TaskId": 25106,
            "BaseLocation": "downloads.dell.com",
            "ManifestIdentifier": "ccb8a137-aa0b-4101-8f92-d524b5e27e56",
            "ReleaseIdentifier": "6D09X",
            "ManifestVersion": "19.01.00",
            "ReleaseDate": "2018-12-31 10:15:59.000",
            "LastUpdated": "2019-01-17 12:19:33.440",
            "BundlesCount": 246,
            "PredecessorIdentifier": "74c35fa7-c094-4ab7-97cd-0e716bd92bf3",
            "AssociatedBaselines": [],
            "Repository": {
                "@odata.type": "#UpdateService.Repository",
                "Id": 12,
                "Name": "c1",
                "Description": "",
```

```
                    "Source": "downloads.dell.com",
                    "DomainName": null,
                    "Username": null,
                    "Password": null,
                    "CheckCertificate": false,
                    "RepositoryType": "DELL_ONLINE"
                }
            }
        ]
}
```


The following response payload indicates that there are multiple baselines associated with this catalog.


```
{
    "@odata.context": "/api/$metadata#Collection(UpdateService.Catalogs)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#UpdateService.Catalogs",
            "@odata.id": "/api/UpdateService/Catalogs(22)",
            "Id": 22,
            "Filename": "catalog.xml",
            "SourcePath": "catalog/catalog.gz",
            "Status": "Completed",
            "TaskId": 27539,
            "BaseLocation": "ftp.dell.com",
            "ManifestIdentifier": "26920c45-0742-44cd-a2c6-c2f2d2df4037",
            "ReleaseIdentifier": "NH6TJ",
            "ManifestVersion": "18.07.02",
            "ReleaseDate": "2018-07-17 10:10:32.000",
            "LastUpdated": "2018-08-08 18:20:22.981",
            "BundlesCount": 230,
            "PredecessorIdentifier": "a05daa5b-7b98-413c-9000-5806cf1d836a",
            "AssociatedBaselines": [
                {
                    "BaselineId": 10,
                    "BaselineName": "No ID"
                },
                {
                    "BaselineId": 9,
                    "BaselineName": "Single"
                },
                {
                    "BaselineId": 11,
                    "BaselineName": "500"
                },
                {
                    "BaselineId": 6,
                    "BaselineName": "Test"
                },
```

```
                    {
                        "BaselineId": 7,
                        "BaselineName": "Test 100"
                    },
                    {
                        "BaselineId": 8,
                        "BaselineName": "Test 2500+"
                    }
                ],
                "Repository": {
                    "@odata.type": "#UpdateService.Repository",
                    "Id": 12,
                    "Name": "Dell",
                    "Description": "",
                    "Source": "downloads.dell.com",
                    "DomainName": null,
                    "Username": null,
                    "Password": null,
                    "CheckCertificate": false,
                    "RepositoryType": "DELL_ONLINE"
                }
            }
        ]
}
```


3.1 Create firmware baseline

Create a firmware baseline to associate a catalog with one or more devices or groups of devices.

POST /api/UpdateService/Baselines

Description: This method creates a baseline

Result Codes:

HTTP Response Code	Description
201	Created a baseline successfully (for devices or device groups)

Privilege: Users with BASELINE_MANAGEMENT privilege can create baselines. This privilege maps to the Admin user role.

Request:

Attribute Name	Type	Description	Required
Name	String	Name of the baseline	Yes
Description	String	Description of the baseline	No

Attribute Name	Type	Description	Required
CatalogId	Integer	ID of the catalog—Users must enumerate all catalogs and match the “Name” of the repository with the input provided while creating the catalog	Yes
RepositoryId	Integer	ID of the repository – Derived from the catalog response	Yes
DowngradeEnabled	Boolean	Indicates if the firmware can be downgraded	No
Is64Bit	Boolean	This must always be set to true—The size of the DUP files used is 64-bits.	No
Targets:	Array	The DeviceID, if the baseline is being created for devices or, the GroupID, if the baseline is being created for a group of devices. DeviceIDs can be determined through /api/DeviceService/Devices	Yes
Id	Integer	GroupIDs can be determined through /api/GroupService/Groups	
Type:			
Id	String	ID for the device type – DeviceType IDs can be determined through /API/DeviceService/DeviceType	
Name	String	Type of the target (DEVICE or GROUP)	Yes

Example: Request Payload

```
{
    "Name":"Dell Online - Individual Devices",
    "Description":"",
    "CatalogId":24,
    "RepositoryId":14,
    "Targets":[
       {
          "Id":25004,
          "Type":{
             "Id":0,
             "Name":"DEVICE"
          }
       }
     ]
 }
```

HTTP status: 201 (Indicates successful baseline creation)

Example: Response Payload

```
{
    "Id": 0,
    "Name": "Dell Online - Individual Devices",
    "Description": "",
    "LastRun": null,
```

```
    "CatalogId": 24,
    "RepositoryId": 14,
    "RepositoryName": null,
    "RepositoryType": null,
    "DowngradeEnabled": false,
    "Is64Bit": false,
    "TaskId": 25113,
    "ComplianceSummary": null,
    "Targets": [
        {
            "Id": 25004,
            "Type": {
                "Id": 0,
                "Name": "DEVICE"
            }
        }
    ],
    "DeviceComplianceReports": []
}
```


Note: Creation of baselines is an asynchronous process – Until the job to compute the compliance report for the baseline is completed, the baseline ID remains 0 and some of the baseline properties default to "null".

GET /api/UpdateService/Baselines

Description: This method lets you view a baseline.

Result Codes:

HTTP Response Code	Description
200	View a previously created baseline

Privilege: VIEW

Response:

Attribute Name	Type	Description
Name	String	Name of the baseline
Description	String	Description of the baseline
CatalogId	Integer	ID of the catalog
RepositoryId	Integer	ID of the repository
TaskId	Integer	Identifier of task which created the baseline
RepositoryName	String	Name of the repository

Attribute Name	Type	Description
RepositoryType	String	Type of the repository
LastRun	String	Date and time when the baseline was last run.
DowngradeEnabled	Boolean	Field to indicate if downgrade is possible
Is64Bit	Boolean	Always set to “true”
Target:		
Id	Integer	Identifier of device / group that this baseline targets
Type:		
Id	Integer	Device Type ID
Name	String	DeviceType derived “Name”
ComplianceSummary:		
ComplianceStatus	String	Indicates if the upgrade is critical
NumberOfCritical	Integer	Number of critical devices
NumberOfWarning	Integer	Number of warning devices
NumberOfNormal	Integer	Number of normal devices
NumberOfDowngrade	Integer	Number of downgrade devices
Link to Device compliance report	Link	Displays the link to the device compliance report

Example: Response Payload


```
{
    "@odata.context": "/api/$metadata#Collection(UpdateService.Baselines)",
    "@odata.count": 6,
    "value": [
        {
            "@odata.type": "#UpdateService.Baselines",
            "@odata.id": "/api/UpdateService/Baselines(8)",
            "Id": 8,
            "Name": "Test 2500+",
            "Description": "",
            "CatalogId": 22,
            "RepositoryId": 12,
            "TaskId": 27734,
            "RepositoryName": "Dell",
            "RepositoryType": "DELL_ONLINE",
            "LastRun": "2018-08-08 21:22:36.858",
            "DowngradeEnabled": true,
            "Is64Bit": true,
            "Targets": [
                {
                    "Id": 1010,
                    "Type": {
```

```
                        "Id": 6000,
                        "Name": "GROUP"
                    }
                }
            ],
            "ComplianceSummary": {
                "ComplianceStatus": "CRITICAL",
                "NumberOfCritical": 7761,
                "NumberOfWarning": 1,
                "NumberOfNormal": 12,
                "NumberOfDowngrade": 1
            },
            "DeviceComplianceReports@odata.navigationLink":
"/api/UpdateService/Baselines(8)/DeviceComplianceReports"
        }
    ]
}
```


3.2 View firmware baseline compliance

The firmware baseline compliance report displays the compliance of the selected devices or device groups. The compliance specifies if an action is required. When more than one device is associated with a baseline, the status of the device with the least compliance level to the baseline is indicated as the compliance level of that baseline. For example, if many devices are associated with a firmware baseline and the compliance level of one device in this group is **Critical** while the compliance level of others may be **OK** or **Downgrade**, the compliance level of the baseline is indicated as **Critical**. However, you can view the firmware compliance of individual devices associated with a firmware baseline to either upgrade or downgrade the firmware version on that device.

GET /api/UpdateService/Baselines(8)/DeviceComplianceReports

Description: This method returns the compliance report for the specified baseline

Result Codes:

HTTP Response Code	Description
200	View the compliance report for a particular baseline

Privilege: VIEW

Response:

Attribute Name	Type	Description
DeviceId	Integer	Identifier of the device in the appliance.
ServiceTag	String	Service Tag of the device
DeviceModel	String	Model of the device
DeviceTypeName	String	Type of device, such as server/chassis.

Attribute Name	Type	Description
DeviceName	String	Name of the device
FirmwareStatus	String	Indicates if firmware is compliant or not
ComplianceStatus	String	Indicates the compliance status
DeviceTypeId	Integer	Numeric value for the device type such as server(1000)
RebootRequired	Boolean	Indicates if a reboot is required to make the device compliant.
Link		Link to the component compliance report, identifying the software components on the device and if they match up against the catalog.

Example: Response Payload (OpenManage Enterprise 3.1 and later):

```
{
 {"@odata.context":
"/api/$metadata#Collection(UpdateService.ComponentComplianceReport)",
    "@odata.count": 3,
    "value": [
        {
            "@odata.type": "#UpdateService.ComponentComplianceReport",
            "@odata.id":
"/api/UpdateService/Baselines(6)/DeviceComplianceReports(1)/ComponentComplianceR
eports(3)",
            "Id": 3,
         "DeviceId": 10867,
"ServiceTag": "7T2W1V1",
"DeviceModel": "PowerEdgeT320",
"DeviceTypeName": "SERVER",
"DeviceName": "idrac-7T2W1V1",
"FirmwareStatus": "NonCompliant",
"ComplianceStatus": "CRITICAL",
"DeviceTypeId": 1000,
"RebootRequired": true,
"DeviceFirmwareUpdateCapable": true,
"DeviceUserFirmwareUpdateCapable": true,
“ComponentComplianceReports": [
{
"@odata.type": "#UpdateService.ComponentComplianceReport",
"Id": 35,
"Version": "16.01.08",
"CurrentVersion": "0",
"Path": "FOLDER03532998M/2/T320_Drivers-for-OS-
Deployment_Application_1HPN9_WN64_16.01.08_A00.EXE",
"Name": "OS Drivers Pack",
"Criticality": "Optional",
"UniqueIdentifier": "1HPN9LW64769c0bf7ebf2f62442730b81c55c0605",
"TargetIdentifier": "18981",
"UpdateAction": "UPGRADE",
```

```
"SourceName": "DCIM:INSTALLED#802__DriverPack.Embedded.1:LC.Embedded.1",
"PrerequisiteInfo": "",
"ImpactAssessment": "",
"Uri":
"http://www.dell.com/support/home/us/en/19/Drivers/DriversDetails?driverId=1HPN9
",
"RebootRequired": false,
"ComplianceStatus": "WARNING",
"ComplianceDependencies": []
},
{
"@odata.type": "#UpdateService.ComponentComplianceReport",
"Id": 39,
"Version": "4247A1",
"CurrentVersion": "0",
"Path":
"FOLDER03035031M/1/Diagnostics_Application_D5TM2_WN64_4247A1_4247.2.EXE",
"Name": "Enterprise UEFI Diagnostics",
"Criticality": "Optional",
"UniqueIdentifier": "D5TM2LW644967bc792c1f9fc58995fe1fbeae3d75",
"TargetIdentifier": "25806",
"UpdateAction": "UPGRADE",
"SourceName": "DCIM:INSTALLED#802__Diagnostics.Embedded.1:LC.Embedded.1",
"PrerequisiteInfo": "",
"ImpactAssessment": "",
"Uri":
"http://www.dell.com/support/home/us/en/19/Drivers/DriversDetails?driverId=D5TM2
",
"RebootRequired": false,
"ComplianceStatus": "WARNING",
"ComplianceDependencies": []
},
},
    …    }
}
```


Note: In OME 3.1 and later, the user will have no navigation link. The component compliance report is expanded in the DeviceComplianceReports data itself.

Example: Response Payload (OpenManage Enterprise 3.0):


```
{
    "@odata.context":
"/api/$metadata#Collection(UpdateService.DeviceComplianceReport)",
    "@odata.count": 1,
    "value": [
        {
            "@odata.type": "#UpdateService.DeviceComplianceReport",
```

```
            "@odata.id":
"/api/UpdateService/Baselines(6)/DeviceComplianceReports(1)",
            "Id": 1,
            "DeviceId": 25013,
            "ServiceTag": "B5JLMN2",
            "DeviceModel": "PowerEdge C6420",
            "DeviceTypeName": "SERVER",
            "DeviceName": "idrac-B5JLMN2",
            "FirmwareStatus": "Non-Compliant",
            "ComplianceStatus": "CRITICAL",
            "DeviceTypeId": 1000,
            "RebootRequired": true,
            "ComponentComplianceReports@odata.navigationLink":
"/api/UpdateService/Baselines(6)/DeviceComplianceReports(1)/ComponentComplianceR
eports"
        }
    ]
}
```

Note: In OME 3.0, the user must drill down further to check compliance at the component level using the navigation link in the response above. See link highlighted in bold font and used in subsequent calls.

GET /api/UpdateService/Baselines(9)/DeviceComplianceReports(9523)/ComponentComplianceReports

Description: This method returns the component level compliance report for a particular baseline

Result Codes:

HTTP Response Code	Description
200	View the compliance report at the component level

Privilege: VIEW

Response:

Attribute Name	Type	Description
Version	String	Version of the component as available in the catalog
CurrentVersion	String	Current version of the component
Path	String	Relative path of the DUP file for this component
Name	String	User readable name for the component
Criticality	String	Indicates the urgency of the update for the component
UniqueIdentifier	String	Unique id of the component
TargetIdentifier	String	Shorter target identifier of the component
UpdateAction	String	Indicates if the component is upgraded or downgraded

Attribute Name	Type	Description
SourceName	String	String identifier of the software component. It also identifies the location on the device. More information on the software inventory and source name mappings to devices can be found at /api/DeviceService/Devices(<device id>)/InventoryDetails('deviceSoftware')
PrerequisiteInfo	String	Indicates if there are any prerequisites for updating the component to the new version
ImpactAssessment	String	Similar to criticality, but largely not filled.
Uri	String	Link to the new driver
RebootRequired	Boolean	Indicates if a reboot is required to update the component
ComplianceStatus	String	Compliance status of this component with respect to the catalog.
ComplianceDependencies	String	Intercomponent dependencies if any.

Example: Response Payload

The output below indicates there are 11 components. The detailed response of only one component—SAS RAID firmware, is shown here.


```
{"@odata.context":
"/api/$metadata#Collection(UpdateService.ComponentComplianceReport)",
    "@odata.count": 11,
    "value": [
        {
            "@odata.type": "#UpdateService.ComponentComplianceReport",
            "@odata.id":
"/api/UpdateService/Baselines(6)/DeviceComplianceReports(1)/ComponentComplianceR
eports(11)",
            "Id": 11,
            "Version": "25.5.5.0005",
            "CurrentVersion": "25.5.3.0005",
            "Path": "FOLDER04905010M/1/SAS-
RAID_Firmware_F675Y_WN64_25.5.5.0005_A13.EXE",
            "Name": "PERC H730P Mini Monolithic",
            "Criticality": "Recommended",
            "UniqueIdentifier": "F675YLW64b8e0efd84d9e9525d9655a64ba68309a",
            "TargetIdentifier": "101560",
            "UpdateAction": "UPGRADE",
            "SourceName": "DCIM:INSTALLED#301_C_RAID.Mezzanine.1-1",
            "PrerequisiteInfo": "",
            "ImpactAssessment": "",
            "Uri":
"http://www.dell.com/support/home/us/en/19/Drivers/DriversDetails?driverId=F675Y
",
            "RebootRequired": true,
```

```
            "ComplianceStatus": "CRITICAL",
            "ComplianceDependencies": []
        },
    …
    }
}
```


3.3 Update the firmware

Update firmware by using the firmware baseline compliance report

In this case, you can update the firmware for one or more components on one or more devices, by referring to the compliance report for the devices against the catalog.

Update using firmware baseline compliance report

POST /api/JobService/Jobs

Description: This method initiates the firmware update task by using the baseline compliance report

Result Codes:

HTTP Response Code	Description
201	Successfully initiated the firmware update task and created a job id that can be used to track to completion.

Privilege: Users with JOB_MANAGEMENT and DEVICE_UPDATE privilege (role wise, this maps to Admin/Device Manager users).

Request:

Attribute Name	Type	Description	Required
Id	Integer	Default to 0	No
JobName	String	Name of the update job	Yes
JobDescription	String	Description of the update job	No
Schedule	String	When to run	Yes
State	String	State for the update job	Yes
JobType			
ID	Integer	Use 5 for update tasks	
Name	String	Type of the task (Update_Task)	Yes
ComplianceReportID	String	Baseline ID (this is mistakenly named compliance report ID)	Yes
RepositoryID	String	Repository ID derived from enumerating catalogs	Yes

Attribute Name	Type	Description	Required
catalogID	String	Catalog ID derived from enumerating catalogs	Yes
operationName	String	Type of operation (install firmware)	Yes
complianceUpdate	Boolean	Indicates if this update must match compliance	Yes
signVerify	Boolean	Indicates if the DUP signature must be verified	Yes
stagingValue	Boolean	If “true” stages updates and runs them on the next reboot of the system	
Targets			
ID	Integer	Identifier of the device to be updated	Yes
Data	String	String identifying component to be updated (seen from the component compliance report ID “SourceName” field) For multiple components separate each value by a semi-colon	Yes
TargetType			
ID	Integer	Numeric device type ID	Yes
Name	String	Device/Group	Yes

Example: Request Payload

```
{
    "Id":0,
    "JobName":"Update Firmware: single device, single component",
    "JobDescription":"Firmware Update Job: for single device, single component",
    "Schedule":"startNow",
    "State":"Enabled",
    "JobType":{
       "Id":5,
       "Name":"Update_Task"
    },
    "Params":[
       {
          "JobId":0,
          "Key":"complianceReportId",
          "Value":"6"
       },
       {
          "JobId":0,
```

```
            "Key":"repositoryId",
            "Value":"14"
         },
         {
            "JobId":0,
            "Key":"catalogId",
            "Value":"24"
         },
         {
            "JobId":0,
            "Key":"operationName",
            "Value":"INSTALL_FIRMWARE"
         },
         {
            "JobId":0,
            "Key":"complianceUpdate",
            "Value":"true"
         },
         {
            "JobId":0,
            "Key":"signVerify",
            "Value":"true"
         },
         {
            "JobId":0,
            "Key":"stagingValue",
            "Value":"false"
         }
      ],
      "Targets":[
         {
            "Id":25013,
            "Data":"DCIM:INSTALLED#301_C_RAID.Mezzanine.1-1÷",
            "TargetType":{
               "Id":1000,
               "Name":"DEVICE"
            }
         }
      ]
   }
```


Example: Response Payload


```
{
    "Id": 25125,
    "JobName": "Update Firmware: single device, single component",
    "JobDescription": "Firmware Update Job: for single device, single
component",
    "NextRun": null,
    "LastRun": null,
    "StartTime": null,
```

```
"EndTime": null,
"Schedule": "startNow",
"State": "Enabled",
"CreatedBy": "admin",
"UpdatedBy": null,
"LastRunStatus": {
    "Id": 2200,
    "Name": "NotRun"
},
"JobType": {
    "Id": 5,
    "Name": "Update_Task",
    "Internal": false
},
"JobStatus": {
    "Id": 2080,
    "Name": "New"
},
"Targets": [
    {
        "JobId": 25125,
        "Id": 25013,
        "Data": "DCIM:INSTALLED#301_C_RAID.Mezzanine.1-1;",
        "TargetType": {
            "Id": 1000,
            "Name": "DEVICE"
        }
    }
],
"Params": [
    {
        "JobId": 25125,
        "Key": "complianceReportId",
        "Value": "6"
    },
    {
        "JobId": 25125,
        "Key": "repositoryId",
        "Value": "14"
    },
    {
        "JobId": 25125,
        "Key": "catalogId",
        "Value": "24"
    },
    {
        "JobId": 25125,
        "Key": "operationName",
        "Value": "INSTALL_FIRMWARE"
    },
```

```
            {
                "JobId": 25125,
                "Key": "complianceUpdate",
                "Value": "true"
            },
            {
                "JobId": 25125,
                "Key": "signVerify",
                "Value": "true"
            },
            {
                "JobId": 25125,
                "Key": "stagingValue",
                "Value": "false"
            }
        ],
        "Visible": true,
        "Editable": true,
        "Builtin": false
}
```

Monitoring Jobs to completion:

GET on /api/JobService/Jobs to determine job status and monitor to completion. A list of job states is available below

```
{
            "2020": "Scheduled",
            "2030": "Queued",
            "2040": "Starting",
            "2050": "Running",
            "2060": "Completed",
            "2070": "Failed",
            "2090": "Warning",
            "2080": "New",
            "2100": "Aborted",
            "2101": "Paused",
            "2102": "Stopped",
            "2103": "Canceled"
        }
```

If a job fails, run a GET command on /api/JobService/Jobs(<id>)/ExecutionHistories

You can drill down further by parsing the ID returned in the ExecutionHistories response and using that ID to run a GET call on /api/JobService/Jobs(<id>)/ExecutionHistories(<id>)/ExecutionHistoryDetails.

4 Behavioral differences

This section explains the behavioral differences between OpenManage Enterprise and OpenManage Enterprise – Modular after the firmware update operation.

In OpenManage Enterprise – Modular auto-start of the inventory task is not done and it is initiated manually by clicking on “Check Compliance” in the UI

The API workflow has a small change specific to OpenManage Enterprise – Modular. After the firmware update job is completed, find the internal Job exposed when creating the baseline. See the data returned on a GET of the baseline in question to get the job id (bold Black text in the following code).


```
{
    "@odata.context": "/api/$metadata#Collection(UpdateService.Baselines)",
    "@odata.count": 6,
    "value": [
        {
            "@odata.type": "#UpdateService.Baselines",
            "@odata.id": "/api/UpdateService/Baselines(8)",
            "Id": 8,
            "Name": "Test 2500+",
            "Description": "",
            "CatalogId": 22,
            "RepositoryId": 12,
            "TaskId": 27734,
            "RepositoryName": "Dell",
            "RepositoryType": "DELL_ONLINE",
            "LastRun": "2018-08-08 21:22:36.858",
            "DowngradeEnabled": true,
            "Is64Bit": true,
            "Targets": [
                {
                    "Id": 1010,
                    "Type": {
                        "Id": 6000,
                        "Name": "GROUP"
                    }
                }
            ],
            "ComplianceSummary": {
                "ComplianceStatus": "CRITICAL",
                "NumberOfCritical": 7761,
                "NumberOfWarning": 1,
                "NumberOfNormal": 12,
                "NumberOfDowngrade": 1
            },
            "DeviceComplianceReports@odata.navigationLink":
"/api/UpdateService/Baselines(8)/DeviceComplianceReports"
        }
    ]
}
```

The equivalent action to clicking on “Check Compliance” is to issue a POST on /api/JobService/Actions/JobService.RunJobs with the payload specified below

```
{
    "JobIds": [27734]
}
```

Once this completes the inventory will be refreshed and subsequent calls to check the compliance against /api/UpdateService/Baselines(<BaselineID>)/DeviceComplianceReports should succeed and indicate current status.

In OpenManage Enterprise when the firmware update job completes, an inventory task is kicked off automatically – this enables future requests on the baseline compliance to be “current” as data has been refreshed. If all the internal jobs have completed, then the compliance data remains current. If the internal auto-spawned baseline job (after the inventory task completes) is still running, the POST operation to run the same job will fail with an error message indicating that the job may already be running. This needs to be gracefully handled by the application or the script but no impact to compliance data and the data remains current. If the POST operation is kicked off before the internal auto-spawned baseline job, it’s possible that the internal job will fail and reflect an error in the UI though there is no impact to compliance data and it stays current

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

A.1 Related resources

Provide a list of documents and other assets that are referenced in the paper; include other resources that may be helpful.