

RAID Profile

Document Number: DCIM1031
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2011-03-22

Version: 1.1.0



THIS PROFILE IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND. ABSENT A SEPARATE AGREEMENT BETWEEN YOU AND DELL™ WITH REGARD TO FEEDBACK TO DELL ON THIS PROFILE SPECIFICATION, YOU AGREE ANY FEEDBACK YOU PROVIDE TO DELL REGARDING THIS PROFILE SPECIFICATION WILL BE OWNED AND CAN BE FREELY USED BY DELL.

© 2010 Dell Inc. All rights reserved. Reproduction in any manner whatsoever without the express written permission of Dell, Inc. is strictly forbidden. For more information, contact Dell.

Dell and the *DELL* logo are trademarks of Dell Inc. *Microsoft* and *WinRM* are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

CONTENTS

1	Scope	6
2	Normative References.....	6
2.1	Approved References	6
2.2	Other References.....	6
3	Terms and Definitions	6
4	Symbols and Abbreviated Terms	8
5	Synopsis	8
6	Description	9
7	Implementation Requirements	10
7.1	Views.....	10
7.2	Attributes	21
7.3	DCIM_RAIDService	27
7.4	RAID Profile Registration.....	28
8	Methods.....	29
8.1	Method: DCIM_RAIDService.AssignSpare ()	29
8.2	Method: DCIM_RAIDService.ResetConfig ()	30
8.3	Method: DCIM_RAIDService.ClearForeignConfig ()	30
8.4	Method: DCIM_RAIDService.DeleteVirtualDisk ()	31
8.5	Method: DCIM_RAIDService.CreateVirtualDisk ()	32
8.6	Method: DCIM_RAIDService.GetDHSDisks ()	35
8.7	Method: DCIM_RAIDService.GetRAIDLevels ()	36
8.8	Method: DCIM_RAIDService.GetAvailableDisks ()	37
8.9	Method: DCIM_RAIDService.CheckVDValues ()	38
8.10	Method: DCIM_RAIDService.SetControllerKey ()	40
8.11	Method: DCIM_RAIDService.LockVirtualDisk ()	42
8.12	Method: DCIM_RAIDService.CreateTargetedConfigJob()	42
8.13	Method: DCIM_RAIDService.DeletePendingConfiguration ()	44
8.14	Method: DCIM_RAIDService.SetAttribute()	45
8.15	Method: DCIM_RAIDService.SetAttributes()	46
8.16	Method: DCIM_RAIDService.RemoveControllerKey ()	48
8.17	Method: DCIM_RAIDService.EnableControllerEncryption ().....	49
8.18	Method: DCIM_RAIDService.ReKey ().....	51
8.19	Method: DCIM_RAIDService.UnassignSpare ()	52
9	Use Cases.....	53
9.1	Discovery of RAID Profile support	53
9.2	Inventory of RAID controllers in system.....	54
9.3	Get the first RAID controller's information	54
9.4	Inventory of virtual disks in system	54
9.5	Inventory of physical disks in system.....	55
9.6	Apply pending values for a particular RAID configuration	55
9.7	Delete pending values for a particular RAID configuration.....	55
9.8	Clear old configuration from newly added hard drives	55
9.9	Determine available RAID configurations for a given set of disks	56
9.10	Determine available physical disks for a given RAID configuration	56
9.11	Check available virtual disk parameters for a given RAID level and set of physical disks	56
9.12	Create a virtual disk on the system.....	57
9.13	Determine available physical disks to be used as a hotspare	57
9.14	Assign a physical disk as a hotspare	57
9.15	Delete a virtual disk from the system.....	58
9.16	Delete all virtual disks and unassign all hotspares	58
9.17	Encrypt a virtual disk.....	59
	ANNEX A (informative) Related MOF Files	60

Figures

Figure 1 –RAID Profile: Class Diagram	9
---	---

Tables

Table 1 – Related Profiles	8
Table 2 – CIM Elements: RAID Profile	10
Table 3 – DCIM_ControllerView - Operations	11
Table 4 – DCIM_ControllerView - Properties	11
Table 5 – DCIM_EnclosureView - Operations	13
Table 6 – DCIM_EnclosureView - Properties	14
Table 7 – DCIM_VirtualDiskView - Operations	15
Table 8 – DCIM_VirtualDiskView - Properties	16
Table 9 – DCIM_PhysicalDiskView - Operations	18
Table 10 – DCIM_PhysicalDiskView - Properties	19
Table 11 – DCIM_RAIDEnumeration - Operations	21
Table 12 – Class: DCIM_RAIDEnumeration	22
Table 13 – DCIM_RAIDEnumeration Attributes	22
Table 14 – DCIM_RAIDString - Operations	23
Table 15 – Class: DCIM_RAIDString	24
Table 16 – DCIM_RAIDString Attributes	24
Table 17 – DCIM_RAIDInteger - Operations	25
Table 17 – Class: DCIM_RAIDInteger	25
Table 18 – DCIM_RAIDInteger Attributes	26
Table 19 – DCIM_RAIDService – Operations	27
Table 20 – Class: DCIM_RAIDService	28
Table 21 – DCIM_LCRegisteredProfile - Operations	28
Table 22 – Class: CIM_RegisteredProfile	28
Table 23 – DCIM_RAIDService.AssignSpare () Method: Return Code Values	29
Table 24 – DCIM_RAIDService.AssignSpare () Method: Standard Messages	29
Table 25 – DCIM_RAIDService.AssignSpare () Method: Parameters	29
Table 26 – DCIM_RAIDService.ResetConfig () Method: Return Code Values	30
Table 27 – DCIM_RAIDService.ResetConfig () Method: Standard Messages	30
Table 28 – DCIM_RAIDService.ResetConfig () Method: Parameters	30
Table 29 – DCIM_RAIDService.ClearForeignConfig () Method: Return Code Values	31
Table 30 – DCIM_RAIDService.ClearForeignConfig () Method: Standard Messages	31
Table 31 – DCIM_RAIDService.ClearForeignConfig () Method: Parameters	31
Table 32 – DCIM_RAIDService.ClearForeignConfig () Method: Return Code Values	32
Table 33 – DCIM_RAIDService.DeleteVirtualDisk() Method: Standard Messages	32
Table 34 – DCIM_RAIDService.DeleteVirtualDisk () Method: Parameters	32
Table 35 – DCIM_RAIDService.CreateVirtualDisk () Method: VDPProp (Cachecade)	33
Table 36 – DCIM_RAIDService.CreateVirtualDisk () Method: VDPProp	33
Table 37 – DCIM_RAIDService.CreateVirtualDisk () Method: Return Code Values	34
Table 38 – DCIM_RAIDService.CreateVirtualDisk () Method: Standard Messages	34
Table 39 – DCIM_RAIDService.CreateVirtualDisk () Method: Parameters	35
Table 40 – DCIM_RAIDService.GetDHSDisks () Method: Return Code Values	35
Table 41 – DCIM_RAIDService.GetDHSDisks () Method: Standard Messages	35

Table 42 – DCIM_RAIDService.GetDHSDisks () Method: Parameters.....	36
Table 43 – DCIM_RAIDService.GetRAIDLevels () Method: Return Code Values	36
Table 42 – DCIM_RAIDService.GetRAIDLevels () Method: Standard Messages.....	36
Table 44 – DCIM_RAIDService.GetRAIDLevels () Method: Parameters	36
Table 45 – DCIM_RAIDService.GetAvailableDisks () Method: Return Code Values.....	37
Table 46 – DCIM_RAIDService.GetAvailableDisks () Method: Standard Messages	37
Table 47 – DCIM_RAIDService.GetAvailableDisks () Method: Parameters.....	37
Table 48 – DCIM_RAIDService.CheckVDValues() Method:	38
Table 49 – DCIM_RAIDService.CheckVDValues() Method:	39
Table 50 – DCIM_RAIDService.CheckVDValues () Method: Return Code Values	39
Table 51 – DCIM_RAIDService.CheckVDValues () Method: Standard Messages	39
Table 52 – DCIM_RAIDService.CheckVDValues () Method: Parameters.....	40
Table 53 – DCIM_RAIDService.SetControllerKey () Method: Return Code Values	41
Table 54 – DCIM_RAIDService.SetControllerKey () Method: Standard Messages.....	41
Table 55 – DCIM_RAIDService.SetControllerKey () Method: Parameters.....	41
Table 56 – DCIM_RAIDService.LockVirtualDisk () Method: Return Code Values	42
Table 57 – DCIM_RAIDService.LockVirtualDisk () Method: Standard Messages	42
Table 58 – DCIM_RAIDService.LockVirtualDisk () Method: Parameters	42
Table 59 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Return Code Values	43
Table 60 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Standard Messages	43
Table 61 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Parameters	43
Table 62 – DCIM_RAIDService.DeletePendingConfiguration () Method: Return Code Values	44
Table 63 – DCIM_RAIDService.DeletePendingConfiguration () Method: Standard Messages	44
Table 64 – DCIM_RAIDService.DeletePendingConfiguration () Method: Parameters.....	45
Table 65 – DCIM_RAIDService.SetAttribute() Method: Return Code Values.....	45
Table 66 – DCIM_RAIDService.SetAttribute() Method: Standard Messages	45
Table 67 – DCIM_RAIDService.SetAttribute() Method: Parameters.....	46
Table 69 – DCIM_RAIDService.SetAttributes() Method: Return Code Values	47
Table 70 – DCIM_RAIDService.SetAttributes() Method: Standard Messages	47
Table 71 – DCIM_RAIDService.SetAttributes() Method: Parameters.....	47
Table 72 – DCIM_RAIDService. RemoveControllerKey () Method: Return Code Values.....	49
Table 73 – DCIM_RAIDService. RemoveControllerKey () Method: Standard Messages.....	49
Table 74 – DCIM_RAIDService. RemoveControllerKey () Method: Parameters.....	49
Table 75 – DCIM_RAIDService.EnableControllerEncryption () Method: Return Code Values	49
Table 76 – DCIM_RAIDService. EnableControllerEncryption() Method: Standard Messages	49
Table 77 – DCIM_RAIDService. EnableControllerEncryption() Method: Parameters	50
Table 78 – DCIM_RAIDService. ReKey () Method: Return Code Values	51
Table 79 – DCIM_RAIDService. ReKey () Method: Standard Messages	51
Table 80 – DCIM_RAIDService. ReKey () Method: Parameters	51
Table 81 – DCIM_RAIDService.UnassignSpare () Method: Return Code Values	52
Table 82 – DCIM_RAIDService.UnassignSpare () Method: Standard Messages.....	52
Table 83 – DCIM_RAIDService.UnassignSpare () Method: Parameters	53

RAID Profile

1 Scope

The RAID Profile extends the management capabilities of referencing profiles by adding the capability to represent the configuration of RAID storage. The RAID storage is modeled as collections of attributes where there are collections for the storage adaptors, physical disks, logical disks, end enclosures and parent-child relationships between the collections. Additionally, there is a configuration service that contains all the methods used to configure the RAID storage.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved References

DMTF DSP0131, *Profile Registration Profile 1.0.0*

DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

DMTF DSP0226, *Web Services for Management (WS-Management) Specification 1.1.0*

DMTF DSP0227, *WS-Management CIM Binding Specification 1.0.0*

2.2 Other References

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*, <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>

Unified Modeling Language (UML) from the Open Management Group (OMG), <http://www.uml.org>

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

For the purposes of this document, the following terms and definitions apply.

3.1

can

used for statements of possibility and capability, whether material, physical, or causal

3.2

cannot

used for statements of possibility and capability, whether material, physical, or causal

3.3

conditional

indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met

3.4

mandatory

indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.5

may

indicates a course of action permissible within the limits of the document

3.6

need not

indicates a course of action permissible within the limits of the document

3.7

optional

indicates a course of action permissible within the limits of the document

3.8

referencing profile

indicates a profile that owns the definition of this class and can include a reference to this profile in its "Related Profiles" table

3.9

shall

indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.10

shall not

indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.11

should

indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12

should not

indicates that a certain possibility or course of action is deprecated but not prohibited

3.13

ENUMERATE

Refers to WS-MAN `ENUMERATE` operation as described in Section 8.2 of DSP0226_V1.1 and Section 9.1 of DSP0227_V1.0

3.14

GET

Refers to WS-MAN GET operation as defined in Section 7.3 of DSP00226_V1.1 and Section 7.1 of DSP0227_V1.0

3.15

Cachecade

The cachecade feature makes use of highperforming solid state disks (SSDs) as a secondary tier of cache to provide faster reads and to maximize transactional I/O performance.

4 Symbols and Abbreviated Terms

4.1

CIM

Common Information Model

4.2

iDRAC

integrated Dell Remote Access Controller – management controller for blades and monolithic servers

4.3

CMC

Chassis Management Controller – management controller for the modular chassis

5 Synopsis

Profile Name: RAID Profile

Version: 1.1.0

Organization: Dell

CIM Schema Version: 2.2

Central Class: DCIM_RAIDService

Scoping Class: CIM_ComputerSystem

The RAID Profile extends the management capability of the referencing profiles by adding the capability to describe the RAID configuration. DCIM_RAIDService shall be the Central Class. CIM_ComputerSystem shall be the Scoping Class. Instance(s) of DCIM_RAIDService shall be the Central Instance(s). The instance of CIM_ComputerSystem with which the Central Instance is associated through the CIM_HostedService association shall be the Scoping Instance.

Table 1 identifies profiles that are related to this profile.

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration Profile	DMTF	1.0	Mandatory

6 Description

The RAID Profile describes the RAID configuration service and the groups that the service manages. The profile also describes the relationship of the RAID groups to the profile version information.

Figure 1 represents the class schema for the RAID Profile. For simplicity, the prefix CIM_ has been removed from the names of the classes.

The RAID service in a managed system is represented by the instance of DCIM_RAIDService class. Each RAID controller can have three additional view classes populated besides the Controller view class shown. Views are related to devices through the FQDD.

The profile information is represented with the instance of CIM_RegisteredProfile.

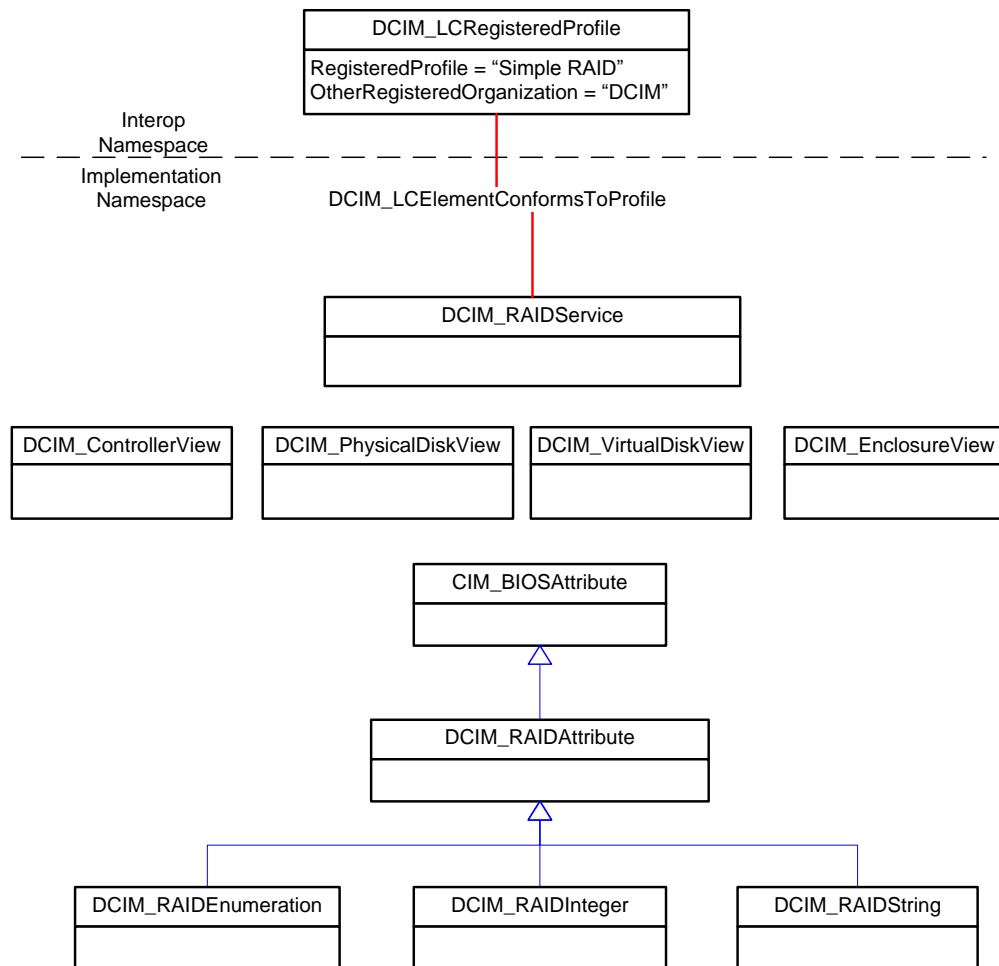


Figure 1 –RAID Profile: Class Diagram

7 Implementation Requirements

Requirements and guidelines for propagating and formulating certain properties of the classes are discussed in this section. Methods are discussed in section 8.

Table 2 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 2. Sections 7.1 (“Implementation Requirements” and “Methods”) may impose additional requirements on these elements.

Table 2 – CIM Elements: RAID Profile

Element Name	Requirement	Description
Classes		
DCIM_RAIDService	Mandatory	See section 7.3
DCIM_ControllerView	Mandatory	See section 7.1.1
DCIM_EnclosureView	Mandatory	See section 7.1.2
DCIM_VirtualDiskView	Mandatory	See section 7.1.3
DCIM_PhysicalDiskView	Mandatory	See section 7.1.4
DCIM_RAIDEnumeration	Mandatory	See section 7.2.1
DCIM_RAIDString	Mandatory	See section 7.2.2
DCIM_RAIDInteger	Mandatory	See section 7.2.3
DCIM_LCElementConformsToProfile	Mandatory	See section 7.4
DCIM_LCRegisteredProfile	Mandatory	See section 7.4
Indications		
None defined in this profile		

7.1 Views

The view classes group object class inventory properties and status properties into a single convenient class. The FQDD property in the view correlates the view properties to a specific device, such as a controller for the controller view.

7.1.1 DCIM_ControllerView - Controller View

This section describes the implementation for the DCIM_ControllerView class. This class shall be instantiated in the Implementation Namespace.

7.1.1.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_ControllerView?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_ControllerView instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_ControllerView?__cimnamespace=root/dcim+InstanceID=<FQDD>”

7.1.1.2 Operations

The following table details the implemented operations on DCIM_ControllerView.

Table 3 – DCIM_ControllerView - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.1.1.3 Properties

The following table details the implemented properties for DCIM_ControllerView instance representing the RAID Controller in a system.

Table 4 – DCIM_ControllerView - Properties

Properties	Data type	Requirement	Additional Requirements
InstanceID	string	Mandatory	The property shall have the value of the FQDD.
FQDD	string	Mandatory	The property shall represent Fully Qualified Device Description (FQDD)
PrimaryStatus	uint32	Mandatory	The property shall represent the status of the device and shall have one of the following values: 0 - Unknown, 1 - OK, 2 - Degraded, 3 - Error.
RollupStatus	uint32	Mandatory	The property shall represent the status of children and shall have one of the following values: 0 - Unknown, 1 - OK, 2 - Degraded, 3 - Error.
ControllerFirmwareVersion	string	Mandatory	The property shall represent the firmware version.
PCISlot	uint8	Mandatory	The property shall represent the associated PCI slot
Bus	string	Mandatory	The property shall represent the PCI Bus
Device	string	Mandatory	The property shall represent the PCI device.
Function	string	Mandatory	The property shall represent the PCI function
PCIVendorID	string	Mandatory	The property shall represent the PCI vendor identifier
PCISubVendorID	string	Mandatory	The property shall represent the PCI sub vendor identifier

Properties	Data type	Requirement	Additional Requirements
PCIDeviceID	string	Mandatory	The property shall represent the PCI device identifier
PCISubDeviceID	string	Mandatory	The property shall represent the PCI sub device identifier
DeviceCardManufacturer	string	Mandatory	The property shall represent the manufacturer name
DeviceCardDataBusWidth	uint8	Mandatory	The property shall represent the bus width and shall have one of the following values: 0 - Unknown, 1- 8x
DeviceCardSlotLength	uint8	Mandatory	The property shall represent the slot length width and shall have one of the following values: 3 - Short, 4 – Long
DeviceCardSlotType	string	Mandatory	The property shall represent the the slot type and shall have one of the following values: “Unknown”, “PCI Express x8”
SecurityStatus	uint32	Mandatory	The property shall represent the controller security configuration information and shall have one of the following values: 0 - Unknown, 1 - Encryption Capable, 2 - Security Key Assigned
ProductName	string	Mandatory	The property shall represent the name of the controller.
SASAddress	string	Mandatory	The property shall represent the provides unique id of controller and shall be in the form of hexadecimal.
EncryptionMode	uint8	Mandatory	The property shall represent the current encryption state on the controller and shall have one of the following values: 0 - None, 1 - Local Key Management, 2 - Dell Key Management, 3 - Pending Dell Key Management
EncryptionCapability	uint8	Mandatory	The property shall represent the EncryptionCapability property details possible encryption states on the controller and shall have one of the following values: 0 - None 1 - Local Key Management

Properties	Data type	Requirement	Additional Requirements
			Capable, 2 - Dell Key Management Capable, 3 - Local Key Management and Dell Key Management Capable
KeyID	string	Mandatory	The property shall represent the KeyId of controller when controller is in Local Key Management mode.
CachecadeCapability	uint8	Mandatory	The property shall represent the controller's support of cachecade virtual disk creation and shall have one of the following values: 0 - Cachecade Virtual Disk not supported 1 – Cachecade Virtual Disk supported.
LastSystemInventoryTime	string	Mandatory	This property provides the last time “ System Inventory Collection on Reboot (CSIOR) ” was performed. The value is represented as “yyyymmddHHMMSS”.
LastUpdateTime	string	Mandatory	This property provides the last time the data was updated. The value is represented as yyyymmddHHMMSS

7.1.2 Enclosure View

This section describes the implementation for the DCIM_EnclosureView class. This class shall be instantiated in the Implementation Namespace.

7.1.2.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_EnclosureView?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_EnclosureView instance shall be:
“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_EnclosureView?__cimnamespace=root/dcim+InstanceID=<FQDD>”

7.1.2.2 Operations

The following table details the implemented operations on DCIM_EnclosureView.

Table 5 – DCIM_EnclosureView - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.1.2.3 Properties

The following table details the implemented properties for DCIM_EnclosureView instance representing the Enclosure in a system.

Table 6 – DCIM_EnclosureView - Properties

Property	Data type	Requirement	Description
InstanceID	string	Mandatory	The property shall have the value of the FQDD.
FQDD	string	Mandatory	Fully Qualified device description (uniquely identifies device)
PrimaryStatus	UInt32	Mandatory	The property shall represent the status of the device and shall have one of the following values: 0 - Unknown, 1 - OK, 2 - Degraded, 3 - Error.
RollupStatus	UInt32	Mandatory	The property shall represent the status of children and shall have one of the following values: 0 - Unknown, 1 - OK, 2 - Degraded, 3 - Error.
Connector	UInt8	Mandatory	The property shall represent the controller port connection.
WiredOrder	UInt8	Mandatory	The property shall represent the order from controller port (0 for backplane).
ServiceTag	string	Mandatory	The property shall have value up to 10 characters
AssetTag	string	Mandatory	The property shall have value up to 10 characters
Version	string	Mandatory	The property shall represent the Enclosure Management Module (EMM) version
SlotCount	UInt8	Mandatory	The property shall represent the number of drive slots.
EMMCount	UInt8	Mandatory	The property shall represent the number Enclosure Management Modules present
PSUCount	UInt8	Mandatory	The property shall represent the number of power supplies present

Property	Data type	Requirement	Description
FanCount	Unit8	Mandatory	The property shall represent the number of fans present.
TempProbeCount	Uint8	Mandatory	The property shall represent the number of temperature probes present
ProductName	string	Mandatory	The property shall represent the marketing name of the enclosure
LastSystemInventoryTime	string	Mandatory	This property provides the last time "System Inventory Collection on Reboot (CSIOR)" was performed. The value is represented as "yyyymmddHHMMSS".
LastUpdateTime	string	Mandatory	This property provides the last time the data was updated. The value is represented as yyyymmddHHMMSS

7.1.3 Virtual Disk View

This section describes the implementation for the DCIM_VirtualDiskView class. This class shall be instantiated in the Implementation Namespace.

7.1.3.1 Resource URIs for WinRM®

The class Resource URI shall be "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_VirtualDiskView?__cimnamespace=root/dcim"

The key property shall be the InstanceID.

The instance Resource URI for DCIM_VirtualDiskView instance shall be: "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_VirtualDiskView?__cimnamespace=root/dcim+InstanceID=<FQDD>"

7.1.3.2 Operations

The following table details the implemented operations on DCIM_VirtualDiskView.

Table 7 – DCIM_VirtualDiskView - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.1.3.3 Properties

The following table details the implemented properties for DCIM_VirtualDiskView instance representing the Virtual Disk in a system.

Table 8 – DCIM_VirtualDiskView - Properties

Properties	Data type	Requirement	Description
InstanceID	string	Mandatory	The property shall have value of the FQDD property.
FQDD	string	Mandatory	The property shall represent the Fully Qualified Device Description that uniquely identifies a device.
PrimaryStatus	Uint32	Mandatory	The property shall represent the status of the device and shall have one of the following values: 0 - Unknown, 1 - OK, 2 - Degraded, 3 - Error.
RAIDStatus	Uint32	Mandatory	The property shall represent the RAID specific status and shall have one of the following values: 0 - Unknown, 1 - Ready, 2 - Online, 3 - Foreign, 4 - Offline, 5 - Blocked, 6 - Failed, 7 – Degraded
WriteCachePolicy	Uint32	Mandatory	The property shall represent the current write policy and shall have one of the following values: 0 - Unknown 1 - Write Through, 2 - Write Back, 3 - Write Back force
ReadCachePolicy	Uint32	Mandatory	The property shall represent the current read policy and shall have one of the following values: 0 - Unknown 16 - No Read Ahead, 32 - Read Ahead, 64 – Adaptive
RAIDTypes	Uint32	Mandatory	The property shall represent the current RAID level and shall have one of the following values: 0 - No RAID 2 - RAID-0 4 - RAID-1 64 - RAID-5 2048 - RAID-10 8192 - RAID-50
SizeInBytes	Uint64	Mandatory	The property shall represent the size of the virtual disk in Bytes.

Properties	Data type	Requirement	Description
StripeSize	UInt32	Mandatory	The property shall represent the current strip size and shall have one of the following values: 0 - Default 1 - 512Bytes, 2 - 1KB 4 - 2KB 8 - 4 KB 16 - 8 kB 32 - 16 KB 64 - 32 KB 128 - 64 KB 256 - 128 KB 512 - 256 KB 1024 - 512 KB 2048 - 1 MB 4096 - 2 MB 8192 - 4 MB 16384 – 8 MB 32768 – 16 MB
Name	string	Mandatory	The property shall represent the virtual disk name
SpanLength	UInt32	Mandatory	The property shall represent the number of physical disks per span
SpanDepth	UInt32	Mandatory	The property shall represent the number of spans in virtual disk.
PhysicalDiskIDs[]	String	Mandatory	The property shall represent the array of physical disk FQDDs.
VirtualDiskTarget ID	UInt32	Mandatory	The property shall represent the virtual disk target number
RemainingRedundancy	UInt16	Mandatory	The property shall represent the remaining redundancy
DiskCachePolicy	UInt32	Mandatory	The property shall represent the policy for physical disks included in the virtual disk and shall have one of the following values: 0 - Unknown 256 - Default, 512 - Enabled, 1024 - Disabled
ObjectStatus	UInt8	Mandatory	The property shall represent the virtual disk configuration state and shall have one of the following values: 0 - Unknown 1 - Current 2 - PendingDelete 3 – PendingCreate
StartingLBAinBlocks	UInt8	Mandatory	The property shall represent the starting logical block address in blocks for virtual disk.

Properties	Data type	Requirement	Description
Cachecade	Uint8	Mandatory	The property shall represent the Cachecade property can have following values and shall have one of the following values: 0 – Not a cachecade Virtual Disk 1 – Cachecade Virtual Disk
LockStatus	Uint8	Mandatory	The property shall represent if this Virtual Disk is locked and shall have one of the following values: 0 – Unlocked 1 – Locked
LastSystemInventoryTime	string	Mandatory	The property shall represent the last time “ System Inventory Collection on Reboot (CSIOR) ” was performed. The value is represented as “yyymmddHHMMSS”.
LastUpdateTime	string	Mandatory	The property shall represent the last time the data was updated. The value is represented as yyymmddHHMMSS

7.1.4 Physical Disk View

This section describes the implementation for the DCIM_PhysicalDiskView class. This class shall be instantiated in the Implementation Namespace.

7.1.4.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_PhysicalDiskView?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_PhysicalDiskView instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_PhysicalDiskView?__cimnamespace=root/dcim+InstanceID=<FQDD>”

7.1.4.2 Operations

The following table details the implemented operations on DCIM_PhysicalDiskView.

Table 9 – DCIM_PhysicalDiskView - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.1.4.3 Properties

The following table details the implemented properties for DCIM_PhysicalDiskView instance representing the Physical Disk in a system.

Table 10 – DCIM_PhysicalDiskView - Properties

Properties	Data type	Requirement	Description
InstanceID	string	Mandatory	The property shall have the same value as the FQDD property.
FQDD	string	Mandatory	The property shall represent the Fully Qualified Device Description that uniquely identifies device.
RAIDStatus	Uint32	Mandatory	The property shall represent the RAID specific status and shall have one of the following values: 0 - Unknown, 1 - Ready, 2 - Online, 3 - Foreign, 4 - Offline, 5 - Blocked, 6 - Failed, 7 – Degraded
PrimaryStatus	Uint32	Mandatory	The property shall represent the status of the device and shall have one of the following values: 0 - Unknown, 1 - OK, 2 - Degraded, 3 - Error.
Connector	Uint16	Mandatory	The property shall represent the controller port that the physical disk is connected to.
Slot	Uint16	Mandatory	The property shall represent the slot where drive is located
SizeInBytes	Uint64	Mandatory	The property shall represent the coerced size of physical disk
Model	string	Mandatory	The property shall represent the model name of the physical disk
Manufacturer	string	Mandatory	The property shall represent the manufacturer of the physical disk.
ManufacturingDay	Uint16	Mandatory	The property shall represent the manufacturing day part of the special notation to identify manufacturing date for the physical disk.
ManufacturingWeek	Uint16	Mandatory	The property shall represent the manufacturing week part of the special notation to identify manufacturing date for the physical disk.
ManufacturingYear	Uint32	Mandatory	The property shall represent the manufacturing year part of the special notation to identify manufacturing date for the physical disk.

Properties	Data type	Requirement	Description
Revision	string	Mandatory	The property shall represent the revision number of physical disk
SerialNumber	string	Mandatory	The property shall represent the serial number of physical disk
BusProtocol	Uint32	Mandatory	The property shall represent the bus protocol and shall have one of the following values: 0 - Unknown, 1 - SCSI, 2 - PATA, 3 - FIBRE, 4 - USB, 5 - SATA, 6 – SAS
HotSpareStatus	Uint16	Mandatory	The property shall represent the hot-spare status and shall have one of the following values: 0 - No, 1 - Dedicated, 2 – Global
PredictiveFailureState	Uint32	Mandatory	The property shall represent the smart alert presence and shall have one of the following values: 0 - Smart Alert Absent, 1 - Smart Alert Present
SecurityState	Uint32	Mandatory	The property shall represent the security state of the physical disk and shall have one of the following values: 0 - Unknown, 1 - Secured, 2 - Locked, 3 – Foreign
MediaType	Uint32	Mandatory	The property shall represent the drive media type and shall have one of the following values: 0 - Magnetic Drive, 1 - Solid State Drive
FreeSizeInBytes	Uint64	Mandatory	The property shall represent the free space available for use for a virtual disk
UsedSizeInBytes	Uint64	Mandatory	The property shall represent the space already consumed by virtual disks
MaxCapableSpeed	Uint32	Mandatory	The property shall represent the speed that disk is capable of and shall have one of the following values: 0 - Unknown, 1 - 1.5GBS,

Properties	Data type	Requirement	Description
			2 - 3GBS, 4 - 6GBS
SASAddress	string	Mandatory	The property shall represent the SAS address of the drive
LastSystemInventoryTime	string	Mandatory	The property shall represent the last time “ System Inventory Collection on Reboot (CSIOR) ” was performed. The value is represented as “yyyymmddHHMMSS”.
LastUpdateTime	string	Mandatory	The property shall represent the last time the data was updated. The value is represented as yyyymmddHHMMSS

7.2 Attributes

This section details the supported attributes for the storage devices. Not all attributes will be available depending on the controller model. Each attribute is separate instance of the attribute class. The FQDD property correlates all the attributes to a device instance. Attributes can be set using the SetAttribute() method.

The RAIDdefaultWritePolicy, RAIDdefaultReadPolicy and DiskCachePolicy attributes are not applicable for Cachecade Virtual Disk.

7.2.1 DCIM_RAIDEnumeration

This section describes the implementation for the DCIM_RAIDEnumeration class.

Each DCIM_RAIDEnumeration instance is logically associated to a DCIM_ControllerView instance or DCIM_EnclosureView instance or DCIM_PhysicalDiskView instance or DCIM_VirtualDiskView instance. The DCIM_RAIDEnumeration. FQDD property value is equal to the FQDD property value of one of the View instance.

This class shall be instantiated in the Implementation Namespace.

7.2.1.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDEnumeration?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_RAIDEnumeration instance shall be:
“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDEnumeration?__cimnamespace=root/dcim+InstanceID=<FQDD>:<AttributeName>”

7.2.1.2 Operations

The following table details the implemented operations on DCIM_RAIDEnumeration.

Table 11 – DCIM_RAIDEnumeration - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI

Operation Name	Requirements	Required Input
Enumerate	Mandatory	Class URI
DCIM_RAIDService.SetAttribute()	Mandatory	See section 8.14.
DCIM_RAIDService.SetAttributes()	Mandatory	See section 8.15.

7.2.1.3 Properties

The following table details the implemented properties for DCIM_RAIDEnumeration instance representing a RAID enumeration attribute. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 12 – Class: DCIM_RAIDEnumeration

Properties	Notes	Additional Requirements
InstanceID	Mandatory	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	The property value shall be from the “AttributeName” column in Table 13.
CurrentValue	Mandatory	The property value shall be one of the values in the “PossibleValues” column at the corresponding row in Table 13.
PendingValue	Mandatory	The property value shall be one of the values in the “PossibleValues” column at the corresponding row in Table 13.
IsReadOnly	Mandatory	The property value shall be from the “IsReadOnly” column in Table 13.
FQDD	Mandatory	FQDD of the device that the attribute belongs to.
PossibleValues	Mandatory	The property value shall be equal to the array of the values in “PossibleValues” column at the corresponding row in Table 13.

The following table describes the requirements for the AttributeName, IsReadOnly and PossibleValues properties. The PossibleValues is an array property represented in the table as comma delimited list.

Table 13 – DCIM_RAIDEnumeration Attributes

AttributeName	Description	IsReadOnly	PossibleValue
RAIDSsupportedRAIDLevels	Supported RAID levels. This attribute relates to the controller device.	TRUE	RAID-0, RAID-1, RAID-5, RAID-10, RAID-50, RAID-60
RAIDSsupportedDiskProt	Supported disk protocol. This attribute relates to the controller device.	TRUE	SAS, SATA
RAIDloadBalancedMode ¹	Load balance mode. This attribute relates to the controller device.	FALSE	Automatic, Disabled
RAIDbatteryLearnMode ¹	Battery learn mode. This attribute relates to the controller device.	FALSE	Automatic, Warn only, Disabled

AttributeName	Description	IsReadOnly	PossibleValue
RAIDccMode ¹	Check consistency mode. This attribute relates to the controller device.	FALSE	Normal , StopOnError
RAIDprMode ¹	Patrol read mode. This attribute relates to the controller device.	FALSE	Automatic, Manual, Disabled
RAIDcopybackMode ¹	Copy back mode. This attribute relates to the controller device.	FALSE	On, On with SMART, Off
RAIDMaxCapableSpeed ¹	Transfer speed of the controller.	TRUE	1_5_GBS , 3_GBS, 6_GBS
RAIDdefaultWritePolicy ¹	Desired write policy of the virtual disk.	FALSE	WriteThrough, WriteBack, WriteBackForce
RAIDdefaultReadPolicy ¹	Desired read policy of the virtual disk	FALSE	NoReadAhead ,ReadAhead, Adaptive
DiskCachePolicy ¹	Disk cache policy for all member disks. This attribute relates to the virtual disk device.	FALSE	Default, Enabled, Disabled

NOTE: 1 – The attribute may not always be present.

7.2.2 DCIM_RAIDString

This section describes the implementation for the DCIM_RAIDString class.

Each DCIM_RAIDString instance is logically associated to a DCIM_ControllerView instance or DCIM_EnclosureView instance or DCIM_PhysicalDiskView instance or DCIM_VirtualDiskView instance. The DCIM_RAIDString.FQDD property value is equal to the FQDD property value of one of the View instance.

This class shall be instantiated in the Implementation Namespace.

7.2.2.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDString?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_RAIDString instance shall be: “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDString?__cimnamespace=root/dcim+InstanceID=<FQDD>:<AttributeName>”

7.2.2.2 Operations

The following table details the implemented operations on DCIM_RAIDString.

Table 14 – DCIM_RAIDString - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_RAIDService.SetAttribute()	Mandatory	See section 8.14.
DCIM_RAIDService.SetAttributes()	Mandatory	See section 8.15.

7.2.2.3 Properties

The following table details the implemented properties for DCIM_RAIDString instance representing a RAID string attribute. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 15 – Class: DCIM_RAIDString

Properties	Notes	Additional Requirements
InstanceID	Mandatory	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	The property value shall be from the “AttributeName” column in Table 16.
CurrentValue	Mandatory	The property value shall be a string with minimum length specified in “MinLength” column and maximum length specified in “MaxLength” column in Table 16.
PendingValue	Mandatory	The property value shall be a string with minimum length specified in “MinLength” column and maximum length specified in “MaxLength” column in Table 16.
IsReadOnly	Mandatory	The property value shall be the value in the “R/RW” column at the corresponding row in Table 16.
FQDD	Mandatory	FQDD of the device that the attribute belongs to.
MinLength	Mandatory	The property value shall be the value in the “MinLength” column at the corresponding row in Table 16.
MaxLength	Mandatory	The property value shall be the value in the “MaxLength” column at the corresponding row in Table 16.

The following table describes possible attributes and the requirements for the AttributeName, IsReadOnly, MinLength, and MaxLength properties.

Table 16 – DCIM_RAIDString Attributes

AttributeName	Description	IsReadOnly	MinLength	MaxLength
RAIDAssetTag	Asset tag of the enclosure.	TRUE	0	12
Name	Virtual disk name	TRUE	0	15

7.2.3 DCIM_RAIDInteger

This section describes the implementation for the DCIM_RAIDInteger class.

Each DCIM_RAIDInteger instance is logically associated to a DCIM_ControllerView instance or DCIM_EnclosureView instance or DCIM_PhysicalDiskView instance or DCIM_VirtualDiskView instance. The DCIM_RAIDString. FQDD property value is equal to the FQDD property value of one of the View instance.

This class shall be instantiated in the Implementation Namespace.

7.2.3.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDInteger?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_RAIDInteger instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDInteger?__cimnamespace=root/dcim+InstanceID= <FQDD>:<AttributeName>”

7.2.3.2 Operations

The following table details the implemented operations on DCIM_RAIDInteger.

Table 17 – DCIM_RAIDInteger - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_RAIDService.SetAttribute()	Mandatory	See section 8.14.
DCIM_RAIDService.SetAttributes()	Mandatory	See section 8.15.

7.2.3.3 Properties

The following table details the implemented properties for DCIM_RAIDInteger instance representing a RAID integer attribute. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 17 – Class: DCIM_RAIDInteger

Properties	Notes	Additional Requirements
InstanceID	Mandatory	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	The property value shall be from the “AttributeName” column in Table 18.
CurrentValue	Mandatory	The property value shall be equal or greater than the value in the “LowerBound”column and equal or less than the value in the “UpperBound” column in Table 18.
PendingValue	Mandatory	The property value shall be equal or greater than the value in the “LowerBound”column and equal or less than the value in the “UpperBound” column in Table 18.
IsReadOnly	Mandatory	The property value shall be the value in the “IsReadOnly” column at the corresponding row in Table 18.
FQDD	Mandatory	FQDD of the device that the attribute belongs to.
LowerBound	Mandatory	The property value shall be the value in the “LowerBound” column at the corresponding row in Table 18.
UpperBound	Mandatory	The property value shall be the value in the “UpperBound” column at the corresponding row in Table 18.

The following table describes the requirements for the AttributeName, IsReadOnly, LowerBound, and UpperBound properties.

Table 18 – DCIM_RAIDInteger Attributes

AttributeName	Description	IsReadOnly	LowerBound	UpperBound
RAIDmaxSupportedVD	Maximum number of Supported Virtual Disks. The attribute is related to the controller device.	TRUE		
RAIDmaxPDsInSpan	Maximum number of physical disks per span. The attribute is related to the controller device.	TRUE		
RAIDmaxSpansInVD	Maximum number of spans allowed in a virtual disk. The attribute is related to the controller device.	TRUE		
RAIDrebuildRate ¹	Rebuild Rate of the Controller. The attribute is related to the controller device.	FALSE	1	100
RAIDccRate ¹	Check consistency rate of the Controller. The Value ranges form 1-100. The attribute is related to the controller device.	FALSE	1	100
RAIDreconstructRate ¹	Reconstruct rate of the Controller. The attribute is related to the controller device.	FALSE	1	100
RAIDbgiRate ¹	Background Initialization Rate of the controller. The attribute is related to the controller device.	FALSE	1	100
RAIDprRate ¹	Patrol Read Rate of the Controller. The attribute is related to the controller device.	TRUE	1	100

NOTE: 1 – The attribute may not always be present.

7.3 DCIM_RAIDService

This section describes the implementation for the DCIM_RAIDService class.

This class shall be instantiated in the Implementation Namespace.

The DCIM_LCElementConformsToProfile association(s)' ManagedElement property shall reference the DCIM_RAIDService instance(s).

7.3.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDService?__cimnamespace=root/dcim”

The key properties shall be the SystemCreationClassName, CreationClassName, SystemName, and Name.

The instance Resource URI for DCIM_RAIDService instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_RAIDService?__cimnamespace=root/dcim+SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+ SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService”

7.3.2 Operations

The following table details the implemented operations on DCIM_RAIDService.

Table 19 – DCIM_RAIDService – Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
Invoke	Mandatory	Instance URI

7.3.3 Properties

The following table details the implemented properties for DCIM_RAIDService instance representing a system in a system. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 20 – Class: DCIM_RAIDService

Properties and Methods	Requirement	Description
SystemCreationClassName	Mandatory	The property value shall be “DCIM_ComputerSystem”.
CreationClassName	Mandatory	The property value shall be “DCIM_RAIDService”.
SystemName	Mandatory	The property value shall be “DCIM:ComputerSystem”.
Name	Mandatory	The property value shall be “DCIM:RAIDService”

7.4 RAID Profile Registration

This section describes the implementation for the DCIM_LCRegisteredProfile class.

This class shall be instantiated in the Interop Namespace.

The DCIM_ElementConformsToProfile association(s)' ConformantStandard property shall reference the DCIM_LCRegisteredProfile instance.

7.4.1 Resource URIs for WinRM®

The class Resource URI shall be "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_RegisteredProfile?__cimnamespace=root/interop"

The key property shall be the InstanceID property.

The instance Resource URI shall be: “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LCRegisteredProfile?__cimnamespace=root/interop+InstanceID=DCIM:SimpleRAID:1.0.0”

7.4.2 Operations

The following table details the implemented operations on DCIM_SystemView.

Table 21 – DCIM_LCRegisteredProfile - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.4.3 Properties

The following table details the implemented properties for DCIM_LCRegisteredProfile instance representing RAID Profile implementation. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirements”, or shall be implemented according to the requirements in the corresponding column “Additional Requirements”.

Table 22 – Class: CIM_RegisteredProfile

Properties	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Simple RAID”.

RegisteredVersion	Mandatory	This property shall have a value of "1.1.0".
RegisteredOrganization	Mandatory	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	Mandatory	This property shall match "DCIM"

8 Methods

This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM elements defined by this profile

8.1 Method: DCIM_RAIDService.AssignSpare ()

The AssignSpare() method is used to assign a physical disk as a dedicated hot spare for a virtual disk, or as a global hot spare.

Table 23 – DCIM_RAIDService.AssignSpare () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 24 – DCIM_RAIDService.AssignSpare () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing CIM method parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR009	Physical disk FQDD did not identify a valid physical disk for the operation
STOR017	Virtual Disk provided is not valid for the operation

Table 25 – DCIM_RAIDService.AssignSpare () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Physical Disk)
IN	VirtualDiskArray[]	String	Array of ElementName(s) where each ElementName identifies a different virtual disk.
OUT	RebootRequired	string	A value of "Yes" means a reboot is required to set this value, and a value of "No" means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

Qualifiers	Name	Type	Description/Values
OUT	MessageArguments []	string	Substitution variables for dynamic error messages

8.2 Method: DCIM_RAIDService.ResetConfig ()

The ResetConfig () method is used to delete all virtual disks and unassign all hot spare physical disks. **All data on the existing virtual disks will be lost!**

Table 26 – DCIM_RAIDService.ResetConfig () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 27 – DCIM_RAIDService.ResetConfig () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure

Table 28 – DCIM_RAIDService.ResetConfig () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments []	string	Substitution variables for dynamic error messages

8.3 Method: DCIM_RAIDService.ClearForeignConfig ()

The ClearForeignConfig () method is used to prepare any foreign physical disks for inclusion in the local configuration.

Table 29 – DCIM_RAIDService.ClearForeignConfig () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 30 – DCIM_RAIDService.ClearForeignConfig () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR018	No foreign drives detected

Table 31 – DCIM_RAIDService.ClearForeignConfig () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments []	string	Substitution variables for dynamic error messages

8.4 Method: DCIM_RAIDService.DeleteVirtualDisk ()

The DeleteVirtualDisk () method is used to delete a single virtual disk from the targeted controller. The successful execution of this method results in the marking of this virtual disk for deletion. The ObjectStatus property in the Virtual Disk view will have the value “PendingDelete”. The Virtual disk will not be deleted until a configuration job is scheduled and the system is rebooted.

Table 32 – DCIM_RAIDService.ClearForeignConfig () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 33 – DCIM_RAIDService.DeleteVirtualDisk() Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR017	Virtual Disk provided is not valid for the operation

Table 34 – DCIM_RAIDService.DeleteVirtualDisk () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Virtual disk)
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments []	string	Substitution variables for dynamic error messages

8.5 Method: DCIM_RAIDService.CreateVirtualDisk ()

The CreateVirtualDisk () method is used to create a single virtual disk on the targeted controller. The successful execution of this method results in a pending but not yet created virtual disk. The ObjectStatus property in the Virtual Disk view will have the value “PendingCreate”. The virtual disk will not be created until a configuration job has been scheduled and the system is rebooted. Upon creation of the virtual disk the FQDD of the formerly pending virtual disk will change.

This method also supports creation of sliced virtual disk. A sliced virtual disk is created, if CreateVirtualDisk () Size input parameter value is less than total size of the set of physical disks. Additional sliced virtual disk can be created using the same set of physical disks and same raid level that was used to create the first virtual disk. If the set of physical disks already has sliced virtual disks, then the CheckVDValues () method should be used on that set of physical disks to find the exact value for

StartingLBA. This value should be used as the StartingLBA parameter value of the CreateVirtualDisk () method.

This CreateVirtualDisk () method is also used to create a Cachecade Virtual Disk on the targeted controller. This method internally creates a RAID-0 virtual disk. The creation process is same as explained earlier. In this scenario, CreateVirtualDisk () method only takes VDPPropNameArray-VDPPropValueArray pairs mentioned in following table.

Table 35 – DCIM_RAIDService.CreateVirtualDisk () Method: VDPProp (Cachecade)

<i>VDPPropNameArray values</i>	<i>VDPPropValueArray Value Description</i>
Cachecade	The valid input value is 1. (required)
VirtualDiskName	Name (optional)

Table 36 – DCIM_RAIDService.CreateVirtualDisk () Method: VDPProp

<i>VDPPropNameArray values</i>	<i>VDPPropValueArray Value Description</i>
Size	Size, in MB, of the virtual disk.(optional)
RAIDLevel	The new RAID level, such as 0, 1, 5, or 6..(required)
SpanDepth	Number of spans in virtual disk.(required for multispans)
SpanLength	Number of disks per span.(required for multispans)
StripeSize	8k, 16k, 32k ... (optional)
ReadPolicy	(optional)
WritePolicy	(optional)
DiskCachePolicy	(optional)
VirtualDiskName	Name (optional)
Initialize	0 –Fast (Only option)
StartingLBA	Starting logical block address of virtual disks in blocks. If 0xFFFFFFFFFFFFFFFF, startingLBA is calculated programmatically. The value can be in hexadecimal or decimal format. e.g In hexadecimal format 0xFFFF e.g In decimal format 65535,

Table 37 – DCIM_RAIDService.CreateVirtualDisk () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 38 – DCIM_RAIDService.CreateVirtualDisk () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR010	RAID level not supported on controller
STOR011	Stripe size not supported on controller
STOR009	Physical disk FQDD did not identify a valid physical disk for the operation
STOR013	One or more Storage device(s) not in a state where the operation can be completed
STOR009	Physical disk provided is not valid for the operation
STOR015	Maximum virtual disks allowed for this controller has been reached
STOR016	Disks provided are too small to create Virtual Disk of this size
STOR043	Physical Disk is part of Virtual Disk that is not Secondary Raid Level 0
STOR044	All Physical Disks specified are not part of the same disk group
STOR045	Physical Disks have holes, StartingLBA and Size parameters are required to create a Virtual Disk
STOR046	Invalid StartingLBA and/or Size
STOR051	StartingLBA and Size combination goes beyond Physical Disk size
STOR052	Unsupported number of Virtual Disks on a controller or disk group
STOR054	Controller is not cachecade capable.

Table 39 – DCIM_RAIDService.CreateVirtualDisk () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (controller)
IN, REQ	PDArray[]	String	Array of FQDDs where each FQDD identifies a physical disk..
IN, REQ	VDPropNameArray[]	String	Indexed array of Virtual Disk property names with relative values contained in VDPropValueArray parameter.
IN, REQ	VDPropValueArray[]	String	Indexed array of Virtual Disk property values relative to VDPropValueName parameter.
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	NewVirtualDisk	DCIM_VirtualDiskView REF	Reference to new virtual disk
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.6 Method: DCIM_RAIDService.GetDHSDisks ()

The GetDHSDisks () method is used to determine possible choices of drives to be a dedicated hot spare for the identified Virtual disk. GetDHSDisks() returns success if it has evaluated the physical disks for potential hot spares, the PDArray return list can be empty if no physical disks are suitable hot spares.

Table 40 – DCIM_RAIDService.GetDHSDisks () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 41 – DCIM_RAIDService.GetDHSDisks () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR017	Virtual Disk provided is not valid for the operation

Table 42 – DCIM_RAIDService.GetDHSDisks () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Virtual disk)
OUT	PDArray[]	String	Array of FQDDs where each identifies a physical disk
OUT	MessageID	String	Error MessageID is returned If the method fails to execute
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute
OUT	MessageArguments []	String	Substitution variables for dynamic error messages

8.7 Method: DCIM_RAIDService.GetRAIDLevels ()

The GetRAIDLevels() method is used to determine possible choices RAID Levels to create virtual disks. If the list of Physical disks is not provided, this method will operate on all connected disks.

Table 43 – DCIM_RAIDService.GetRAIDLevels () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 42 – DCIM_RAIDService.GetRAIDLevels () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure

Table 44 – DCIM_RAIDService.GetRAIDLevels () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN, REQ	DiskType	Uinit32	0- Include all Types, 1- Include Magnetic only, 2 Include Solid State Only
IN, REQ	Diskprotocol	Uinit32	0 - Include all protocols, 1- Include Sata, 2 Include SAS
IN	DiskEncrypt	Uinit32	0 – Include FDE (encryption capable) and Non encryption capable disks 1 – Include FDE only, include only non FDE disks, 2- Include only non FDE disks

Qualifiers	Name	Type	Description/Values
IN	PDArray[]	String	Array of FQDD(s) identifies physical disk(s)..
OUT	VDRAIDEnumArray[]	String	Indexed array of Virtual Disk RAID level enum values .
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.8 Method: DCIM_RAIDService.GetAvailableDisks ()

The GetAvailableDisks () method is used to determine possible choices of drives to create virtual disks.

Table 45 – DCIM_RAIDService.GetAvailableDisks () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 46 – DCIM_RAIDService.GetAvailableDisks () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure

Table 47 – DCIM_RAIDService.GetAvailableDisks () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN, REQ	DiskType	UInt32	0 - Include all Types, 1- Include Magnetic only, 2 Include Solid State Only
IN, REQ	Diskprotocol	UInt32	0 - Include all protocols, 1- Include Sata, 2 Include SAS
IN	DiskEncrypt	UInt32	0 – Include FDE (encryption capable) and Non encryption capable disks 1 – Include FDE only, include only non FDE disks, 2- Include only non FDE disks
IN	RaidLevel	UInt32	

Qualifiers	Name	Type	Description/Values
OUT	PDArray[]	String	Array of FQDD(s) identifies physical disk(s)..
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments []	String	Substitution variables for dynamic error messages

8.9 Method: DCIM_RAIDService.CheckVDValues ()

The CheckVDValues() method is used to determine possible sizes of Virtual disk as well default settings, given a RAID level and set of disks. The VDPropArray is filled in with Size and other values for a successful execution of the method. If the SpanDepth is not provided a default value of 2 will be used for RAID levels 10, 50 and 60. For certain numbers of disks, such as nine or fifteen, it may be necessary for the user to provide another SpanDepth.

Table 48 – DCIM_RAIDService.CheckVDValues() Method:

<i>VDPropNameArrayIn values</i>	<i>VDPropValueArrayIn Value Description</i>
Size	Size, in MB, of the virtual disk.(optional)
SpanDepth	Number of spans in virtual disk (Required for multispans raidlevel.) The default value is two for Multispans Raidlevels and one for basic RaidLevels
RAIDLevel	Refer to RAIDLevel Values and ValueMaps from DCIM_VirtualDiskView MOF.(required)
StartingLBA	Starting logical block address of virtual disks in 512 byte blocks. If input value is 0xFFFFFFFFFFFFFFFF or 18446744073709551615, startingLBA is calculated programmatically.

Table 49 – DCIM_RAIDService.CheckVDValues() Method:

VDPropNameArrayOut values	VDPropValueArrayOut Value Description
SizeInBytes	If Input Parameter “Size” is not specified or is specified as zero, then “SizeInBytes” returns the maximum allowed size of the Virtual Disk. If the input parameter “Size” is non-zero, SizeInBytes is same as Size.
RAIDLevel	Refer to RAIDLevel Values and ValueMaps from DCIM_VirtualDiskView MOF.
SpanDepth	Number of spans in virtual disk.
SpanLength	Number of disks per span.
StripeSize	Refer to Virtual Disk view
ReadPolicy	Refer to Virtual Disk view
WritePolicy	Refer to Virtual Disk view
DiskCachePolicy	Refer to Virtual Disk view
Name	Virtual Disk Name.
StartingLBA	Starting logical Block address in 512 byte blocks of the Virtual Disk.

Table 50 – DCIM_RAIDService.CheckVDValues () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 51 – DCIM_RAIDService.CheckVDValues () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR010	RAID level not supported on controller
STOR011	Stripe size not supported on controller
STOR009	Physical disk FQDD did not identify a valid physical disk for the operation
STOR013	One or more Storage device(s) not in a state where the operation can be completed
STOR017	Virtual Disk provided is not valid for the operation
STOR035	Not enough Storage objects or Storage objects in incorrect state for this operation

MessageID (OUT parameter)	Message
STOR043	Physical Disk is part of Virtual Disk that is not Secondary Raid Level 0
STOR044	All Physical Disks specified are not part of the same disk group
STOR045	Physical Disks have holes, StartingLBA and Size parameters are required to create a Virtual Disk
STOR046	Invalid StartingLBA and/or Size
STOR051	StartingLBA and Size combination goes beyond Physical Disk size
STOR052	Unsupported number of Virtual Disks on a controller or disk group

Table 52 – DCIM_RAIDService.CheckVDValues () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN, REQ	PDArray[]	String	Array of FQDD(s) identifies physical disk(s).
IN, REQ	VDPropNameArrayIn[]	String	Indexed array of Virtual Disk property names with relative values contained in VDPropValueArray parameter.
IN, REQ	VDPropValueArrayIn[]	String	Indexed array of Virtual Disk property values relative to VDPropValueName parameter.
OUT	VDPropNameArray[]	String	Indexed array of Virtual Disk property names with relative values contained in VDPropValueArray parameter.
OUT	VDPropValueArray[]	String	Indexed array of Virtual Disk property values relative to VDPropValueName parameter.
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.10 Method: DCIM_RAIDService.SetControllerKey ()

The SetControllerKey() method sets the key on controllers and sets controller in Local key Management (LKM) mode that support encryption of the drives.

Table 53 – DCIM_RAIDService.SetControllerKey () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 54 – DCIM_RAIDService.SetControllerKey () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR020	Controller Key is already present
STOR022	Controller is not security capable
STOR038	Invalid parameter value for Keyid

Table 55 – DCIM_RAIDService.SetControllerKey () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN, REQ	Key	String	Key-Passcode. The Key can be maximum 32 characters long, and should have one character from each of the following set. <ol style="list-style-type: none"> 1. Upper Case 2. Lower Case 3. Number 4. Special Character. The special characters in the following set need to be passed as mentioned below. <ol style="list-style-type: none"> 1. & → &amp; 2. < → &lt; 3. > → &gt; 4. “ → &quot; 5. ‘ → &apos;
IN, REQ	Keyid	String	Key Identifier- Describes Key. The Keyid can be maximum 32 characters long and should not have spaces in it.
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

Qualifiers	Name	Type	Description/Values
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.11 Method: DCIM_RAIDService.LockVirtualDisk ()

The LockVirtualDisk() method encrypts the virtual disk.

Table 56 – DCIM_RAIDService.LockVirtualDisk () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 57 – DCIM_RAIDService.LockVirtualDisk () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR021	Controller Key is not present

Table 58 – DCIM_RAIDService.LockVirtualDisk () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Virtual Disk)
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.12 Method: DCIM_RAIDService.CreateTargetedConfigJob()

The CreateTargetedConfigJob() method is used to apply the pending values set by previous RAIDService methods.

CreateTargetedConfigJob method supports the following optional input parameters

1. RebootJobType: when provided in the input parameters, creates a specific reboot job to “PowerCycle” or “Graceful Reboot without forced shutdown” or “Graceful Reboot with forced shutdown”. This parameter only creates the RebootJob and does not schedule it.
2. ScheduledStartTime: When provided in the input parameters, schedules the “configuration job” and the optional “reboot job” at the specified start time. A special value of “TIME_NOW” schedules the job(s) immediately.
3. UntilTime: This parameter has a dependency on “ScheduledStartTime”, together “ScheduledStartTime” and “UntilTime” define a time window for scheduling the job(s). Once scheduled, jobs will be executed within the time window.
4. If CreateTargetedConfigJob method is executed without the 3 optional parameters discussed above, then configuration job is created but not scheduled. However, this configuration job can be scheduled later using the DCIM_JobService.SetupJobQueue () method from the “Job Control Profile”.
DCIM_JobService.SetupJobQueue () can be executed to schedule several configuration jobs including the reboot job. Refer to “Job Control Profile” for more details.

Table 59 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 60 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR026	Configuration Job not Created, there are no pending Configuration changes
STOR024	Configuration already committed, cannot commit until previous commit succeeds or is cancelled
STOR023	Configuration already committed, cannot set configuration

Table 61 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN	RebootJobType	uint16	Shall contain the requested reboot type: 1 - PowerCycle 2 - Graceful Reboot without forced shutdown 3 - Graceful Reboot with forced shutdown.

Qualifiers	Name	Type	Description/Values
IN	ScheduledStartTime	string	Start time for the job execution in format: yyyyymmddhhmmss. The string "TIME_NOW" means immediate.
IN	UntilTime	string	End time for the job execution in format: yyyyymmddhhmmss. : If this parameter is not NULL, then ScheduledStartTime parameter shall also be specified.
OUT	Job	CIM_ConcreteJob REF	Reference to the newly created pending value application job.
OUT	MessageID	string	Error Message ID- can be used to index into Dell Message registry files
OUT	Message	string	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	string	Substitution variables for dynamic error messages

8.13 Method: DCIM_RAIDService.DeletePendingConfiguration ()

The DeletePendingConfiguration() method cancels the pending configuration changes made before the configuration job is created with CreateTargetedConfigJob(). This method only operates on the pending changes prior to CreateTargetedConfigJob() being called. After the Configuration job is created the pending changes can only be canceled by calling CancelJob() in the Job Control profile.

Table 62 – DCIM_RAIDService.DeletePendingConfiguration () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 63 – DCIM_RAIDService.DeletePendingConfiguration () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR025	Configuration already committed, cannot delete pending configuration

Table 64 – DCIM_RAIDService.DeletePendingConfiguration () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.14 Method: DCIM_RAIDService.SetAttribute()

The SetAttribute() method is used to set or change the value of a RAID attribute.

Invocation of the SetAttribute() method shall change the value of the CIM_Attribute.CurrentValue or CIM_Attribute.PendingValue property to the value specified by the AttributeValue parameter if the CIM_AttributeValue.IsReadOnly property is FALSE. Invocation of this method when the CIM_AttributeValue.IsReadOnly property is TRUE shall result in no change to the value of the CIM_AttributeValue.CurrentValue property. The results of changing this value is described with the SetResult parameter.

Return code values for the SetAttribute() method are specified in Table 65. Standard messages are specified in Table 66, and parameters are specified in Table 67. Invoking the SetAttribute() method multiple times can result in the earlier requests being overwritten or lost.

Table 65 – DCIM_RAIDService.SetAttribute() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred

Implementation of standard messages is optional. Standard messages defined for this method are described in Table 66.

Table 66 – DCIM_RAIDService.SetAttribute() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR006	General failure
STOR007	Resource Allocation Failure
STOR039	Mismatch in AttributeName and AttributeValue count
STOR037	Missing required parameter <Parameter Name>
STOR038	Invalid parameter value for <Parameter Name>
STOR040	Invalid Attribute Name <Attribute Name>
STOR041	Invalid Attribute Value for Attribute Name <Attribute Name>

MessageID(OUT parameter)	Message
STOR042	Unsupported Attribute Value for Attribute Name <Attribute Name>
STOR047	AttributeValue cannot be changed for ReadOnly Attribute Name <Attribute Name>

Table 67 – DCIM_RAIDService.SetAttribute() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of of target device
IN, REQ	AttributeName	String	Shall contain the attribute name representing the attribute to be modified, as specified by Attribute.AttributeName property. The specified attribute shall be unique and already exist.
OUT	SetResult	String	Returns: "Set CurrentValue" when the attributes current value is set. "Set PendingValue" when the attributes pending value is set.
IN, REQ	AttributeValue[]	array of strings	Shall contain a new value to assign to the specified Attribute. If this value is valid, it will be applied to the CurrentValue or PendingValue property of the specified Attribute depending on the system implementation.
OUT	RebootRequired	string	A value of "Yes" means a reboot is required to set this value, and a value of "No" means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments []	String	Substitution variables for dynamic error messages

8.15 Method: DCIM_RAIDService.SetAttributes()

The SetAttributes() method is used to set or change the values of a group of attributes.

Invocation of the SetAttributes() method shall change the values of the CIM_Attribute.CurrentValue or PendingValue properties that correspond to the names specified by the AttributeName parameter and the values specified by the AttributeValue parameter if the respective CIM_Attribute.IsReadOnly property is FALSE. Invocation of this method when the respective CIM_Attribute.IsReadOnly property is TRUE shall result in no change to the corresponding value of the CIM_Attribute.CurrentValue property.

If more than one value is specified for a particular attribute, the AttributeName parameter shall contain multiple identical array entries that represent the attribute name that corresponds to each respective attribute value described by the AttributeValue parameter.

Return code values for the SetAttributes() method are specified in Table 68. Standard messages are specified in Table 69, and parameters are specified in Table 70.

Invoking the SetAttributes() method multiple times can result in the earlier requests being overwritten or lost.

Table 68 – DCIM_RAIDService.SetAttributes() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred

Standard messages defined for this method are described in **Error! Reference source not found..**

Table 69 – DCIM_RAIDService.SetAttributes() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR006	General failure
STOR007	Resource Allocation Failure
STOR039	Mismatch in AttributeName and AttributeValue count
STOR037	Missing required parameter <Parameter Name>
STOR038	Invalid parameter value for <Parameter Name>
STOR040	Invalid Attribute Name <Attribute Name>
STOR041	Invalid Attribute Value for Attribute Name <Attribute Name>
STOR042	Unsupported Attribute Value for Attribute Name <Attribute Name>
STOR047	AttributeValue cannot be changed for ReadOnly Attribute Name <Attribute Name>

Table 70 – DCIM_RAIDService.SetAttributes() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of of target device
IN, REQ	AttributeName[]	String	Shall contain the attribute names representing the attributes to be modified, as specified by Attribute.AttributeName properties. The specified attributes must already exist. The values of Attribute.AttributeName supplied for this parameter shall be unique within the scope of the instantiating Namespace. The attribute name members of this array must correspond with array members of the values represented by the AttributeValue parameter. If more than one value is specified for a particular attribute, this parameter shall contain multiple identical array entries describing the attribute name that corresponds with each respective attribute value specified by the AttributeValue parameter.

Qualifiers	Name	Type	Description/Values
OUT	SetResult[]	String	SetResult returns the results of invoking this method for each specified attribute value. Each array element of SetResult contains the result of setting the new value that corresponds with each respective attribute value specified by the AttributeValue parameter. "Set CurrentValue" when the attributes current value is set. "Set PendingValue" when the attributes pending value is set.
IN, REQ	AttributeValue[]	String	Shall contain new values to assign to the attributes specified in the AttributeName parameter. The attribute value members of this array must correspond with the array members of the names represented by the AttributeName parameter. If more than one value is specified for a particular attribute, this parameter shall contain an entry for each attribute value. A value of NULL indicates the factory default values for the Attribute is requested. If this value is valid, it will be applied to the CurrentValue or PendingValue property of the specified Attribute depending on the system implementation and any requirements for a system restart. The result of applying the values are described in the corresponding array entries of the SetResult parameter.
OUT	RebootRequired	String	A value of "Yes" means a reboot is required to set this value, and a value of "No" means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.16 Method: DCIM_RAIDService.RemoveControllerKey ()

The RemoveControllerKey () method erases the encryption key on controller. All encrypted drives are also erased as result of this operation.

Table 71 – DCIM_RAIDService.RemoveControllerKey () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 72 – DCIM_RAIDService.RemoveControllerKey () Method: Standard Messages

MessageID(OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR021	Controller Key is not present
STOR022	Controller is not security capable

Table 73 – DCIM_RAIDService.RemoveControllerKey () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of of target device (Controller)
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.17 Method: DCIM_RAIDService.EnableControllerEncryption ()

The EnableControllerEncryption () method sets either Local Key Management (LKM) or Dell Key Management (DKM) on controllers that support encryption of the of drives.

Table 74 – DCIM_RAIDService.EnableControllerEncryption () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 75 – DCIM_RAIDService.EnableControllerEncryption() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR0003	Missing parameter

MessageID(OUT parameter)	Message
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR019	Provided passphrase is not valid
STOR022	Controller is not security capable
STOR038	Invalid parameter value for Keyid
STOR020	Controller Key is already present

Table 76 – DCIM_RAIDService.EnableControllerEncryption() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of of target device (Controller)
IN, REQ	Mode	Uint16	Mode of the controller 1 - Local Key Management (LKM) 2 – Dell Key Management (DKM)
IN	Key	String	Key – Passcode. This parameter is required if the Mode = LKM. The Key can be maximum 32 characters long, and should have one character from each of the following set. <ol style="list-style-type: none"> 1. Upper Case 2. Lower Case 3. Number 4. Special Character. The special characters in the following set need to be passed as mentioned below. <ol style="list-style-type: none"> 1. & → &amp; 2. < → &lt; 3. > → &gt; 4. “ → &quot; 5. ‘ → &apos;
IN	Keyid	String	Key Identifier- Describes Key. This parameter is required if the Mode = LKM. The Keyid can be maximum 32 characters long and should not have spaces in it.
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.18 Method: DCIM_RAIDService.ReKey ()

The ReKey () method resets the key on the controller that support encryption of the of drives. This method switches the controller mode.

Table 77 – DCIM_RAIDService. ReKey () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 78 – DCIM_RAIDService.ReKey () Method: Standard Messages

MessageID(OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR0019	Provided passphrase is not valid
STOR048	Controller is not Dell Key Management capable
STOR050	Controller is in Dell Key Management mode
STOR053	Controller key not present, controller needs key from Dell Key Management Server
STOR038	Invalid parameter value for Keyid
STOR020	Controller Key is already present

Table 79 – DCIM_RAIDService.ReKey () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of of target device (Controller)
IN, REQ	Mode	Uint16	Mode of the Controller 1 - Local Key Management (LKM) 2 – Dell Key Management (DKM)

Qualifiers	Name	Type	Description/Values
IN	Newkey	String	New controller key. The Key can be maximum 32 characters long, and should have one character from each of the following set. <ol style="list-style-type: none"> 1. Upper Case 2. Lower Case 3. Number 4. Special Character. The special characters in the following set need to be passed as mentioned below. <ol style="list-style-type: none"> 1. & → &amp; 2. < → &lt; 3. > → &gt; 4. " → &quot; 5. ' → &apos;
IN	Oldkey	String	Old controller key.
IN	Keyid	String	Key Identifier- Describes Key. The Keyid can be maximum 32 characters long and should not have spaces in it.
OUT	RebootRequired	string	A value of "Yes" means a reboot is required to set this value, and a value of "No" means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

8.19 Method: DCIM_RAIDService.UnassignSpare ()

The UnassignSpares() method is used to unassign a physical disk as a dedicated hot spare from a virtual disk, or as a global hot spare. After the method executes successfully the physical disk will no longer be a hot spare.

Table 80 – DCIM_RAIDService.UnassignSpare() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred

Table 81 – DCIM_RAIDService.UnassignSpare() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value

MessageID(OUT parameter)	Message
STOR006	General failure
STOR007	Resource Allocation Failure
STOR009	Physical disk FQDD did not identify a valid physical disk for the operation

Table 82 – DCIM_RAIDService.UnassignSpare() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Physical Disk)
OUT	RebootRequired	string	A value of “Yes” means a reboot is required to set this value, and a value of “No” means a reboot is not required to set this value
OUT	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.
OUT	MessageArguments []	String	Substitution variables for dynamic error messages

9 Use Cases

This section contains use cases for the RAID Profile.

Note that URIs in this section are in form of Resource URIs for WinRM®.

9.1 Discovery of RAID Profile support

Use one of the two procedures below to confirm the existence of RAID Profilesupport

- A) GET the *DCIM_LCRegisteredProfile* instance using an *InstanceID* of DCIM:SimpleRAID:1.0.0.
See section 3.14 for a definition of GET .

Instance URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_LCRegisteredProfile?_cimnamespace=root/interop+InstanceID=DCIM:SimpleRAID:1.0.0

Results for the *InstanceID* of DCIM:SimpleRAID:1.0.0 shown below. If no instance is returned, the profile is not supported.

```
DCIM_LCRegisteredProfile
  AdvertiseTypeDescriptions = WS-Identify, Interop Namespace
  AdvertiseTypes = 1, 1
  InstanceID = DCIM:SimpleRAID:1.0.0
```

OtherRegisteredOrganization = DCIM
RegisteredName = Simple RAID
RegisteredOrganization = 1
RegisteredVersion = 1.1.0

B) ENUMERATE the *CIM_RegisteredProfile* class. See section 3.15 for a definition of ENUMERATE.

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_RegisteredProfile?_cimnamespace=root/interop

Then query the result for the following properties:

RegisteredName = Simple RAID, OtherRegisteredOrganization = DCIM, RegisteredVersion = 1.0.0

9.2 Inventory of RAID controllers in system

ENUMERATE the *DCIM_ControllerView* class to view all available instances of the class

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_ControllerView?_cimnamespace=root/dcim

The instance information of all available RAID controllers will be returned

9.3 Get the first RAID controller's information

The URI for getting particular instance information is deterministic (i.e the *InstanceID* will be unique for each instance)

For the first RAID in the system, the instance URI will be:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_ControllerView?_cimnamespace=root/dcim+InstanceID=RAID.Integrated.1-1

The instance of *DCIM_ControllerView* that contains the information on the first RAID controller will be returned

9.4 Inventory of virtual disks in system

ENUMERATE the *DCIM_VirtualDiskView* class to view all available instances of the class

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_VirtualDiskView?_cimnamespace=root/dcim

The instance information of all available virtual disks will be returned

9.5 Inventory of physical disks in system

ENUMERATE the *DCIM_PhysicalDiskView* class to view all available instances of the class

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_PhysicalDiskView?_cimnamespace=root/dcim

The instance information of all available physical disks will be returned

9.6 Apply pending values for a particular RAID configuration

A) To invoke the *CreateTargetedConfigJob()* method, construct input parameters per Table 61 and use the RAID FQDD from section 7.1.1.3

B) INVOKE *CreateTargetedConfigJob()* method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

C) Allow several minutes for the UEFI to execute the SSIB task, which will be followed by a reboot of the system

D) Query the status of the *jobID* output using the job control profile methods

9.7 Delete pending values for a particular RAID configuration

A) To invoke the *DeletePendingConfiguration()* method, construct input parameters per Table 64 and use the RAID FQDD from section 7.1.1.3

B) INVOKE *DeletePendingConfiguration()* method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

C) If the return parameters indicate success, per Table 62, no further action necessary

9.8 Clear old configuration from newly added hard drives

A) To invoke the *ClearForeignConfig()* method, construct input parameters per Table 31 and use the particular RAID's FQDD

B) INVOKE *ClearForeignConfig()* method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerS

[ystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService](http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService)

- C) If return message indicates success per Table 30, no further action necessary

9.9 Determine available RAID configurations for a given set of disks

- A) To invoke the GetRAIDLevels() method, construct input parameters per Table 44 and use the desired RAID FQDDs from Section 9.8
- B) INVOKE GetRAIDLevels() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) The output will contain the available RAID level configurations for the given physical disk selection

9.10 Determine available physical disks for a given RAID configuration

- A) To invoke the GetAvailableDisks() method, construct input parameters per Table 47 and use the desired RAID FQDDs from Section 9.9
- B) INVOKE GetAvailableDisks() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) The output will contain the available physical disks for the given RAID level

9.11 Check available virtual disk parameters for a given RAID level and set of physical disks

- A) To invoke the CheckVDValues() method, construct input parameters per Table 52 and using physical disk FQDDs from Section 9.10 and a RAID level from Section 9.9
- B) INVOKE CheckVDValues() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) The output will contain the available sizes and default values for the given RAID level and set of physical disks

9.12 Create a virtual disk on the system

- A) Determine applicable RAID configuration from Section 9.9
- B) Select physical disks for RAID configuration from Section 9.10
- C) Check available sizes and default virtual disk parameters in Section 9.10 and 9.11
- D) To invoke the CreateVirtualDisk() method, construct input parameters per Table 39 and use the particular RAID FQDDs from Section 9.10
- E) INVOKE CreateVirtualDisk() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- F) Examine output parameters per Table 39
- G) Apply the pending values (Section 9.6) using the FQDD obtained from Section
- H) Allow several minutes for the UEFI to execute the SSIB task, which will be followed by an automatic reboot of the system
- I) List virtual disk inventory, Section 9.4, to check the status of the creation of the new virtual disk

9.13 Determine available physical disks to be used as a hotspare

- A) To invoke the GetDHSDisks() method, construct input parameters per Table 42 and use the applicable RAID virtual disk FQDD from Section 9.12
- B) INVOKE GetDHSDisks() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) The output will contain the available physical disks for use as a hotspare

9.14 Assign a physical disk as a hotspare

- A) Confirm at least one virtual disk exists, see Section 9.4
- B) Determine potential physical disks to be used as a hotspare from Section 9.13. To invoke the AssignSpare() method, construct input parameters per and use the particular RAID virtual disk FQDD from Section 9.10
- C) INVOKE AssignSpare() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- D) Examine output parameters per Table 42
- E) Apply the pending values (Section 9.6) using the FQDD obtained from Section 9.10
- F) Allow several minutes for the UEFI to execute the SSIB task, which will be followed by an automatic reboot of the system
- G) Confirm the applicable physical disk is a hotspare by listing the physical disks, per Section 9.5, and checking that the *HotSpareStatus* attribute is set to '1'

9.15 Delete a virtual disk from the system

- A) To invoke the DeleteVirtualDisk() method, construct input parameters per Table 34 and use the particular RAID's FQDD from Section 9.2.
- B) INVOKE DeleteVirtualDisk () method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) Examine output parameters per Table 34
- D) Apply the pending values (Section 9.6) using the FQDD obtained from Section 9.2
- E) Allow several minutes for the UEFI to execute the SSIB task, which will be followed by an automatic reboot of the system
- F) Confirm successful execution of the method by listing the virtual disks, per Section 9.4 The deleted virtual disk should not be displayed

9.16 Delete all virtual disks and unassign all hotspares

- A) To invoke the ResetConfig() method, construct input parameters per Table 28 and use the FQDD from Section 9.2
- B) INVOKE ResetConfig() method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) Examine output parameters per Table 28
- D) Apply the pending values (Section 9.6) using the FQDD obtained from Section 9.2

- E) Allow several minutes for the UEFI to execute the SSIB task, which will be followed by an automatic reboot of the system
- F) Confirm successful execution of the method by listing the virtual disks and physical disks, per Section 9.4 and Section 9.5, respectively. No virtual disks should be displayed and all physical disk *HotSpareStatus* attributes should be set to '0'

9.17 Encrypt a virtual disk

- A) To invoke the `SetControllerKey()` method, construct input parameters per Table 55 and use the FQDD from Section 9.2

- B) INVOKE `SetControllerKey()` method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- C) To invoke the `LockVirtualDisk()` method, construct input parameters per Table 58 and use the virtual disk FQDD from Section 9.4

- D) INVOKE `LockVirtualDisk()` method

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_RAIDService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_RAIDService+SystemName=DCIM:ComputerSystem+Name=DCIM:RAIDService

- E) Apply the pending values (Section 9.6) using the FQDD obtained from Section 9.2
- F) Allow several minutes for the UEFI to execute the SSIB task, which will be followed by an automatic reboot of the system

ANNEX A (informative)

Related MOF Files

Dell Tech Center MOF Library:

<http://www.delltechcenter.com/page/DCIM.Library.MOF>

Related Managed Object Format (MOF) files:

DCIM_ControllerView.mof
DCIM_EnclosureView.mof
DCIM_PhysicalDiskView.mof
DCIM_RAIDService.mof
DCIM_VirtualDiskView.mof
DCIM_LCElementConformsToProfile.mof
DCIM_LCRegisteredProfile.mof
DCIM_RAIDEnumeration.mof
DCIM_RAIDInteger.mof
DCIM_RAIDString.mof