RAID Profile

Document Number: DCIM1031
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2012-12-30

Version: 1.3.0

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.

CONTENTS

Figures

Tables

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.

- DMTF DSP0131, *Profile Registration Profile 1.0.0*
- DMTF DSP0226, *Web Services for Management (WS-Management) Specification 1.1.0*
- DMTF DSP0227, *WS-Management CIM Binding Specification 1.0.0*
- *Dell Lifecycle Controller Best Practices Guide 1.0,* *http://en.community.dell.com/techcenter/extras/m/white_papers/20066173.aspx*
- *Dell WSMAN Licenses and Privileges 1.0*
- 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
- 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

Dell Tech Center MOF Library: http://www.delltechcenter.com/page/DCIM.Library.MOF.

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

Conditional – Indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met.

3.2

Mandatory – Indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted.

3.3

May – Indicates a course of action permissible within the limits of the document.

3.4

Optional – Indicates a course of action permissible within the limits of the document.

3.5

can – Used for statements of possibility and capability, whether material, physical, or causal.

3.6

cannot – Used for statements of possibility and capability, whether material, physical, or causal.

3.7

need not – 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

FQDD – Fully Qualified Device Descriptor is used to identify a particular component in a system.

3.14

Interop Namespace root/interop – Interop Namespace is where instrumentation instantiates classes to advertise its capabilities for client discovery.

3.15

Implementation Namespace: root/dcim – Implementation Namespace is where instrumentation instantiates classes relevant to executing core management tasks.

3.16

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

3.17

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.

3.18

Cachecade – The cachecade feature makes use of high-performing solid state disks (SSDs) as a secondary tier of cache to provide faster reads 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

4.4

EMM

Enclosure Management Module

5 Synopsis

Profile Name: RAID Profile

Version: 1.3.0

Organization: Dell

CIM Schema Version: 2.26 Experimental

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	DCIM	1.0	Reference

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.

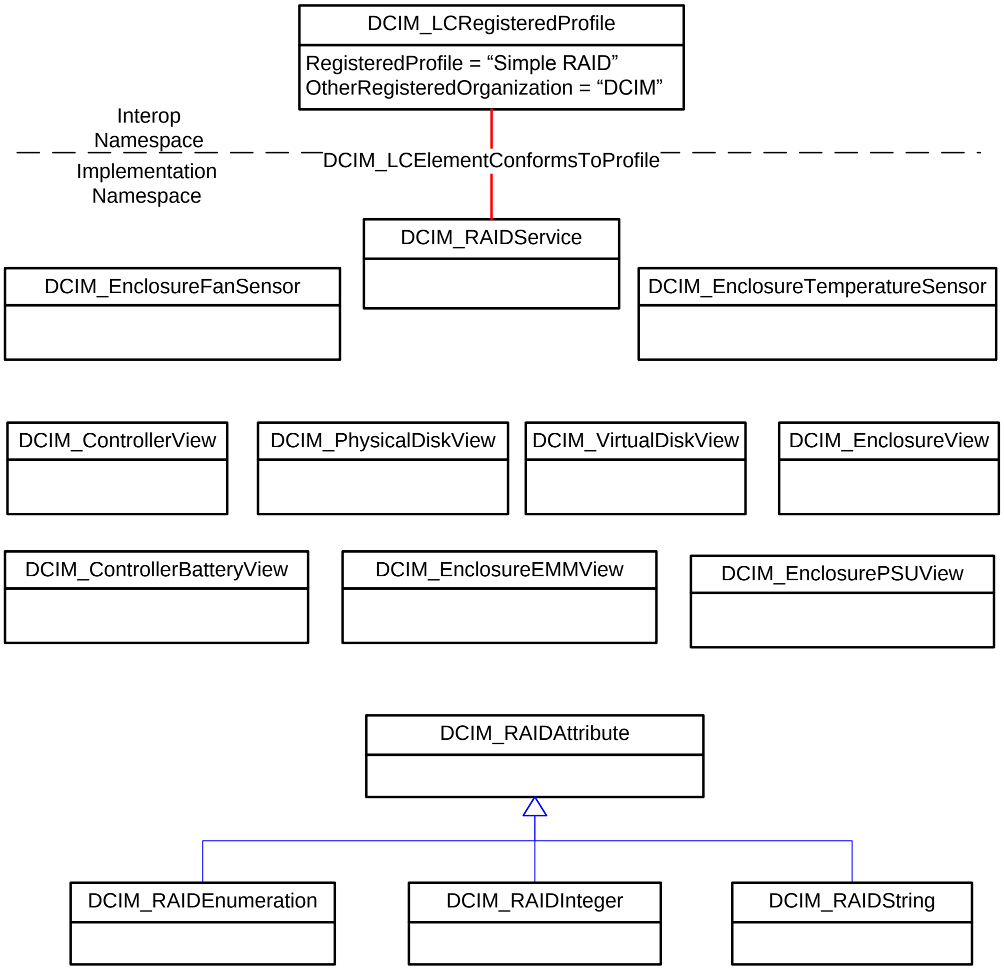

Figure 1 –RAID Profile: Class Diagram

7 Implementation Description

This section describes the requirements and guidelines for implementing RAID profile.

Table 2 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 2.

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_ControllerBatteryView	Mandatory	See section 7.1.5
DCIM_EnclosureEMMView	Mandatory	See section 7.1.6
DCIM_EnclosurePSUView	Mandatory	See section 7.1.7
DCIM_EnclosureFanSensor	Mandatory	See section 7.1.8
DCIM_EnclosureTemperatureSensor	Mandatory	See section 7.1.9
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 View Classes

The view classes represent the properties and status of the storage devices. The FQDD property correlates the view to a specific device such as RAID.Integrated.1-1 for an integrated RAID controller..

7.1.1 Controller View - DCIM_ControllerView

This section describes the implementation for the DCIM_ControllerView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

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 lists 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 Class Properties

The following table lists the implemented properties for DCIM_ControllerView instance representing the RAID Controller in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 4 – DCIM_ControllerView - Properties

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

Property Name	Requirement	Type	Additional Requirements
PCIDeviceID	Mandatory	string	The property shall represent the PCI device identifier.
PCISubDeviceID	Mandatory	string	The property shall represent the PCI sub device identifier.
DeviceCardManufacturer	Mandatory	string	The property shall represent the manufacturer name.
DeviceCardDataBusWidth	Mandatory	uint8	The property shall represent the bus width and shall be one of the following values: • 0 - Unknown • 1- 8x
DeviceCardSlotLength	Mandatory	uint8	The property shall represent the slot length width and shall be one of the following values: • 2 – Unknown • 3 – Short • 4 – Long
DeviceCardSlotType	Mandatory	string	The property shall represent the the slot type and shall be one of the following values: • Unknown • PCI Express x8
SecurityStatus	Mandatory	uint32	The property shall represent the controller security configuration information and shall be one of the following values: • 0 - Unknown • 1 - Encryption Capable • 2 - Security Key Assigned
ProductName	Mandatory	string	The property shall represent the name of the controller.
SASAddress	Mandatory	string	The property shall provide unique ID of the controller and shall be in the form of hexadecimal.
EncryptionMode	Mandatory	uint8	The property shall represent the current encryption state on the controller and shall be one of the following values: • 0 - None • 1 - Local Key Management • 2 - Dell Key Management • 3 - Pending Dell Key Management
EncryptionCapability	Mandatory	uint8	The property shall represent the EncryptionCapability property details possible encryption states on the controller and shall be one of the following values: • 0 - None • 1 - Local Key Management Capable
KeyID	Mandatory	string	The property shall represent the KeyId of controller when controller is in Local Key Management mode.
CachecadeCapability	Mandatory	uint8	The property shall represent the controller's support of cachecade virtual disk creation and shall have one of the following values:

Property Name	Requirement	Type	Additional Requirements
			• 0 - Cachecade Virtual Disk not supported • 1 – Cachecade Virtual Disk supported
SlicedVDCapability	Mandatory	uint8	The property shall represent the controller’s support of sliced virtual disk creation and shall have one of the following values: • 0 - Sliced Virtual Disk not supported • 1 – Sliced Virtual Disk supported
CacheSizeInMB	Mandatory	uint32	This property shall represent the controller cache size in MB.
PatrolReadState	Mandatory	uin8	This property shall represent the current state of the patrol read operation and shall have following values: • 0 – Unknown • 1 – Stopped • 2 – Running
DriverVersion	Mandatory	string	This property shall represent the controller driver version.
MaxPossiblePCILinkSpeed	Mandatory	string	This property shall represent the maximum possible PCI link speed (speed the device is capable of).
MaxAvailablePCILinkSpeed	Mandatory	string	This property shall represent the maximum PCI link speed the controller is set to allow.
LastSystemInventoryTime	Mandatory	string	This property provides the last time “**System Inventory Collection on Reboot (CSIOR**)” was performed. The value is represented as "yyyymmddHHMMSS”.
LastUpdateTime	Mandatory	string	This property provides the last time the data was updated. The value is represented as “yyyymmddHHMMSS”.

7.1.2 Enclosure View - DCIM_EnclosureView

This section describes the implementation for the DCIM_EnclosureView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

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 lists 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 Class Properties

The following table lists the implemented properties for DCIM_EnclosureView instance representing the Enclosure in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 6 – DCIM_EnclosureView - Properties

Property Name	Requirement	Type	Additional Requirements
InstanceID	Mandatory	string	The property shall have the value of the FQDD.
FQDD	Mandatory	string	Fully Qualified device description (uniquely identifies device)
PrimaryStatus	Mandatory	Uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
RollupStatus	Mandatory	Uint32	The property shall represent the status of children and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
Connector	Mandatory	Uint8	The property shall represent the controller port connection.
WiredOrder	Mandatory	Uint8	The property shall represent the connection sequence in a daisy chain of enclosures in the relation to the controller port (0 for backplane).
ServiceTag	Mandatory	string	The property shall contain up to 10 characters.
AssetTag	Mandatory	string	The property shall contain up to 10 characters.
Version	Mandatory	string	The property shall represent the EMM version.

Property Name	Requirement	Type	Additional Requirements
SlotCount	Mandatory	Uint8	The property shall represent the number of drive slots.
EMMCount	Mandatory	Uint8	The property shall represent the number of EMMs present.
PSUCount	Mandatory	Uint8	The property shall represent the number of power supply units present.
FanCount	Mandatory	Unit8	The property shall represent the number of fans present.
TempProbeCount	Mandatory	Uint8	The property shall represent the number of temperature probes present.
ProductName	Mandatory	string	The property shall represent the commercial name of the enclosure.
LastSystemInventoryTime	Mandatory	string	This property shall provide the last time “System Inventory Collection on Reboot (CSIOR)” was performed. The value is represented as "yyyymmddHHMMSS”.
LastUpdateTime	Mandatory	string	This property shall provide the last time the data was updated. The value is represented as yyyymmddHHMMSS

7.1.3 Virtual Disk View - DCIM_VirtualDiskView

This section describes the implementation for the DCIM_VirtualDiskView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

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 Class Properties

The following table lists the implemented properties for DCIM_VirtualDiskView instance representing the Virtual Disk in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 8 – DCIM_VirtualDiskView - Properties

Property Name	Requirement	Type	Description
InstanceID	Mandatory	string	The property shall have value of the FQDD property.
FQDD	Mandatory	string	The property shall represent the Fully Qualified Device Description that uniquely identifies a device.
BusProtocol	Mandatory	Uint32	The property shall represent the bus protocol and shall be one of the following values: • 0 - Unknown • 1 - SCSI • 2 - PATA • 3 - FIBRE • 4 - USB • 5 - SATA • 6 – SAS
Cachecade	Mandatory	Uint8	The property shall represent the Cachecade property can have following values and shall be one of the following values: • 0 – Not a cachecade Virtual Disk • 1 – Cachecade Virtual Disk
DiskCachePolicy	Mandatory	Uint32	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
LockStatus	Mandatory	Uint8	The property shall represent if this Virtual Disk is locked and shall be one of the following values: • 0 – Unlocked • 1 – Locked

Property Name	Requirement	Type	Description
MediaType	Mandatory	Uint32	The property shall represent the drive media type and shall have one of the following values: • 0 – Unknown • 1 – Hard Disk Drive • 2 – Solid State Drive
Name	Mandatory	string	The property shall represent the virtual disk name.
ObjectStatus	Mandatory	Uint8	The property shall represent the virtual disk configuration state and shall be one of the following values: • 0 – Current • 1 – Pending • 2 – Current Virtual Disk Pending Delete • 3 – Pending Create
OperationName	Mandatory	String	This property shall represent the operation that is running on a virtual disk in background. If no operation is running, the value shall be “None”.
OperationPercentComplete	Mandatory	Uint8	This property shall represent the percentage of completion of the operation that is represented by the OperationName property.
PendingOperations	Mandatory	Uint8	This property shall represent the pending operations on the virtual disk. The property shall have one of the following values: • 0 - None • 1 - Fast Init • 2 - Current Virtual Disk Pending Delete • 3 - Pending Create
PhysicalDiskIDs[]	Mandatory	String	The property shall represent the array of physical disk FQDDs that includes dedicated hot spare physical disk FQDDs as well.
PrimaryStatus	Mandatory	Uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error

Property Name	Requirement	Type	Description
RAIDStatus	Mandatory	Uint32	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 • 8 – Non-RAID
RAIDTypes	Mandatory	Uint32	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
ReadCachePolicy	Mandatory	Uint32	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
RemainingRedundancy	Mandatory	Uint16	The property shall represent the remaining redundancy
RollupStatus	Mandatory	Uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
SizeInBytes	Mandatory	Uint64	The property shall represent the size of the virtual disk in Bytes.
SpanDepth	Mandatory	Uint32	The property shall represent the number of spans in virtual disk.
SpanLength	Mandatory	Uint32	The property shall represent the number of physical disks per span
StartingLBAinBlocks	Mandatory	Uint8	The property shall represent the starting logical block address in blocks for virtual disk.

Property Name	Requirement	Type	Description
StripeSize	Mandatory	Uint32	The property shall represent the current strip size and shall be one of the following values: • 0 - Deafult • 1 – 512 Bytes • 2 - 1 KB • 4 - 2 KB • 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
VirtualDiskTarget ID	Mandatory	Uint32	The property shall represent the virtual disk target number
WriteCachePolicy	Mandatory	Uint32	The property shall represent the current write policy and shall be one of the following values: • 0 - Unknown • 1 - Write Through, • 2 - Write Back, • 3 - Write Back force
LastSystemInventoryTime	Mandatory	string	The property shall represent the last time "**System Inventory Collection on Reboot (CSIOR)"** was performed. The value is represented as "yyyymmddHHMMSS".
LastUpdateTime	Mandatory	string	The property shall represent the last time the data was updated. The value is represented as yyyymmddHHMMSS

7.1.4 Physical Disk View - DCIM_PhysicalDiskView

This section describes the implementation for the DCIM_PhysicalDiskView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

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 Class Properties

The following table lists the implemented properties for DCIM_PhysicalDiskView instance representing the Physical Disk in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 10 – DCIM_PhysicalDiskView - Properties

Property Name	Requirement	Type	Description
InstanceID	Mandatory	string	The property shall have the same value as the FQDD property.
FQDD	Mandatory	string	The property shall represent the Fully Qualified Device Description that uniquely identifies the device.
BusProtocol	Mandatory	Uint32	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
Connector	Mandatory	Uint16	The property shall represent the controller port that the physical disk is connected to.

Property Name	Requirement	Type	Description
DriveFormFactor	Mandatory	Uint8	This property shall represent the physical disk form factor and shall be one of the following values: • 0 - Unknown • 1 - 1.8 inch • 2 - 2.5 inch • 3 - 3.5 inch
FreeSizeInBytes	Mandatory	Uint64	The property shall represent the free space available for a virtual disk
HotSpareStatus	Mandatory	Uint16	The property shall represent the hot-spare status and shall be one of the following values: • 0 - No • 1 - Dedicated • 2 – Global
Manufacturer	Mandatory	string	The property shall represent the manufacturer of the physical disk.
ManufacturingDay	Mandatory	Uint16	The property shall represent the physical disk's manufacturing days from the beginning of the manufacturing week, where weeks run from Saturday to Friday.
ManufacturingWeek	Mandatory	Uint16	The property shall represent the manufacturing fiscal weeks from the first Saturday of July in the manufacturing year.
ManufacturingYear	Mandatory	Uint32	The property shall represent the physical disk's manufacturing fiscal year beginning on the first Saturday of July. Note that the calendar year lags the fiscal year.
MaxCapableSpeed	Mandatory	Uint32	The property shall represent the data transfer speed that the disk is capable of and shall be one of the following values: • 0 - Unknown • 1 - 1.5 GBPS • 2 - 3 GBPS • 4 - 6 GBPS
MediaType	Mandatory	Uint32	The property shall represent the drive media type and shall be one of the following values: • 0 – Hard Disk Drive • 1 – Solid State Drive

Property Name	**Requirement**	**Type**	**Description**
Model	Mandatory	string	The property shall represent the model name of the physical disk.
OperationName	Mandatory	String	This property shall represent the background operation that is running on a virtual disk. If no operation is running, the value shall be “None”.
OperationPercentComplete	Mandatory	Uint8	This property shall represent the percentage completion of the operation that is represented by the OperationName property.
PPID	Mandatory	String	The property shall represent the Part Piece Identification (PPID) value for the physical disk.
PredictiveFailureState	Mandatory	Uint32	The property shall represent the smart alert presence and shall be one of the following values: • 0 - Smart Alert Absent • 1 - Smart Alert Present
PrimaryStatus	Mandatory	Uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
RAIDStatus	Mandatory	Uint32	The property shall represent the RAID specific status and shall be one of the following values: • 0 - Unknown • 1 - Ready • 2 - Online • 3 - Foreign • 4 - Offline • 5 - Blocked • 6 - Failed • 7 – Degraded
Revision	Mandatory	string	The property shall represent the revision number of physical disk.

Property Name	**Requirement**	**Type**	**Description**
RollupStatus			The property shall contain one of the following values: - 0(Unknown) - 1(OK) - 2(Degraded) - 3(Error). RollupStatus provides a high level status value, intended to align with Red-Yellow-Green type representation of status.
SASAddress	Mandatory	string	The property shall represent the SAS address of the drive.
SecurityState	Mandatory	Uint32	The property shall represent the security state of the phycials disk and shall be one of the following values: - 0 - Unknown - 1 - Secured - 2 - Locked - 3 – Foreign
SerialNumber	Mandatory	string	The property shall represent the serial number of physical disk.
SizeInBytes	Mandatory	Uint64	The property shall represent the coerced (no configuration data) size of the physical disk.
Slot	Mandatory	Uint16	The property shall represent the slot where drive is located.
SupportedEncryptionTypes[]	Mandatory	Stirng	This property shall represent the supported encryption types on the physical disk. The possible values is FDE (Full Drive Encryption)
UsedSizeInBytes	Mandatory	Uint64	The property shall represent the space already consumed by virtual disks
LastSystemInventoryTime	Mandatory	string	The property shall represent the last time "**System Inventory Collection on Reboot (CSIOR)"** was performed. The value is represented as "yyyymmddHHMMSS".
LastUpdateTime	Mandatory	string	The property shall represent the last time the data was updated. The value is represented as yyyymmddHHMMSS

7.1.5 Controller Battery View - DCIM_ControllerBatteryView

This section describes the implementation for the DCIM_ControllerBatteryView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

7.1.5.1 Resource URIs for WinRM®

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

The key property shall be the InstanceID.

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

7.1.5.2 Operations

The following table lists the implemented operations on DCIM_ControllerBatteryView.

Table 11 – DCIM_ControllerBatteryView - Operations

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

7.1.5.3 Class Properties

The following table lists the implemented properties for DCIM_ControllerBatteryView instance representing the RAID Controller Battery in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 12 – DCIM_ControllerBatteryView - Properties

Property Name	Requirement	Type	Additional Requirements
InstanceID	Mandatory	string	The property shall have the value of the FQDD.
FQDD	Mandatory	string	The property shall represent Fully Qualified Device Description (FQDD)
PrimaryStatus	Mandatory	uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
RAIDState	Mandatory	Uint16	The property shall represent the status of battery and shall be one of the following values: • 0 - Unknown • 1 - Ready • 6 - Failed • 7 - Degraded • 9 - Missing • 10 - Charging • 12 - Below Threshold

Property Name	Requirement	Type	Additional Requirements
PredictiveCapacity	Mandatory	Uint32	The property shall be following values: 0 – Unknown 1 – Ready 6 - Failed

7.1.6 Enclosure Mangement Module View - DCIM_EnclosureEMMView

This section describes the implementation for the DCIM_EnclosureEMMView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

7.1.6.1 Resource URIs for WinRM®

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

The key property shall be the InstanceID.

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

7.1.6.2 Operations

The following table lists the implemented operations on DCIM_EnclosureEMMView.

Table 13 – DCIM_EnclosureEMMView - Operations

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

7.1.6.3 Class Properties

The following table lists the implemented properties for DCIM_EnclosureEMMView instance representing the EMM in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 14 – DCIM_EnclosureEMMView - Properties

Property Name	Requirement	Type	Additional Requirements
InstanceID	Mandatory	string	The property shall have the value of the FQDD.
FQDD	Mandatory	string	The property shall represent Fully Qualified Device Description (FQDD)
PrimaryStatus	Mandatory	uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded

Property Name	Requirement	Type	Additional Requirements
			• 3 - Error
PartNumber	Mandatory	String	The property shall represent the EMM part number.
Revision	Mandatory	String	The property shall represent the version of the EMM firmware.

7.1.7 Enclosure Power Supply Unit View - DCIM_EnclosurePSUView

This section describes the implementation for the DCIM_EnclosurePSUView class. This class shall be instantiated in the Implementation Namespace: root/dcim.

7.1.7.1 Resource URIs for WinRM®

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

The key property shall be the InstanceID.

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

7.1.7.2 Operations

The following table lists the implemented operations on DCIM_EnclosurePSUView.

Table 15 – DCIM_EnclosurePSUView - Operations

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

7.1.7.3 Class Properties

The following table lists the implemented properties for DCIM_EnclosurePSUView instance representing the Enclosure Power Supply Unit in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 16 – DCIM_EnclosurePSUView - Properties

Property Name	Requirement	Type	Additional Requirements
InstanceID	Mandatory	String	The property shall have the value of the FQDD.
FQDD	Mandatory	String	The property shall represent Fully Qualified Device Description (FQDD)
PrimaryStatus	Mandatory	uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown

Property Name	Requirement	Type	Additional Requirements
			• 1 - OK • 2 - Degraded • 3 - Error
PartNumber	Mandatory	String	The property shall represent the enclosure power supply unit part number.

7.1.8 Enclosure Fan Sensor - DCIM_EnclosureFanSensor

This section describes the implementation for the DCIM_EnclosureFanSensor class. This class shall be instantiated in the Implementation Namespace: root/dcim.

7.1.8.1 Resource URIs for WinRM®

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

The key property shall be the SystemCreationClassName, SystemName, CreationClassName and DeviceID.

The instance Resource URI for DCIM_EnclosureFanSensor instance shall be:
"http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_EnclosureFanSensor?__cimnamespace=root/dcim+SystemCreationClassName= DCIM_ComputerSystem+SystemName= DCIM:ComputerSystem+CreationClassName= DCIM_EnclosureFanSensor+DeviceID=<FQDD>"

7.1.8.2 Operations

The following table lists the implemented operations on DCIM_EnclosureFanSensor.

Table 17 – DCIM_EnclosureFanSensor - Operations

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

7.1.8.3 Class Properties

The following table lists the implemented properties for DCIM_EnclosureFanSensor instance representing the Enclosure Fan Sensor in a system. The "Requirements" column shall denote whether the property is implemented (for requirement definitions, see section 3). The "Additional Requirements" column shall denote either possible values for the property, or requirements on the value formulation.

Table 18– DCIM_EnclosureFanSensor - Properties

Property Name	Requirement	Type	Additional Requirements
SystemCreationClassName	Mandatory	String	The property value shall be "DCIM_ComputerSytem".
SystemName	Mandatory	String	The property value shall be "DCIM:ComputerSystem"

Property Name	Requirement	Type	Additional Requirements
CreationClassName	Mandatory	String	The property value shall be “DCIM_EnclosureFanSensor”
DeviceID	Mandatory	String	The property shall have the sensor FQDD value.
FQDD	Mandatory	String	The property shall represent Fully Qualified Device Description (FQDD)
BaseUnits	Mandatory	uint16	The property value shall be 19 (RPM).
CurrentReading	Mandatory	sint32	The present value indicated by the sensor. The property value shall be in rpm.
ElementName	Mandatory	String	The property shall have value “EnclosureFanSensor”.
PrimaryStatus	Mandatory	uint32	The property shall represent the status of the device and shall be one of the following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
RateUnits	Mandatory	uint16	The property value shall be 4. (Per Minute)
SensorType	Mandatory	Uint16	The property value shall be 5. (Tachometer)
UnitModifier	Mandatory	sint32	The property shall have the value 0 denoting that the CurrentReading property value need not be multiplied by the UnitModifier property value.
SettableThresholds[]	Mandatory	uint16	An array representing the writable thresholds supported by sensor. The property shall be NULL because this sensor is a read-only sensor.
SupportedThresholds[]	Mandatory	uint16	The array property shall be NULL.
UpperThresholdCritical	Mandatory	sint32	The array property shall be NULL.
UpperThresholdNonCritical	Mandatory	sint32	The array property shall be NULL.
LowerThresholdCritical	Mandatory	sint32	The array property shall be NULL.
LowerThresholdNonCritical	Mandatory	sint32	The array property shall be NULL.

7.1.9 Enclosure Temperature Sensor - DCIM_EnclosureTemperatureSensor

This section describes the implementation for the DCIM_Enclosure Temperature Sensor class. This class shall be instantiated in the Implementation Namespace: root/dcim.

7.1.9.1 Resource URIs for WinRM®

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

The key property shall be the SystemCreationClassName, SystemName, CreationClassName and DeviceID.

The instance Resource URI for DCIM_EnclosureTemperatureSensor instance shall be: “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_EnclosureTemperatureSensor?__cimnamespace=root/dcim+SystemCreationClassName= DCIM_ComputerSystem+SystemName= DCIM:ComputerSystem+CreationClassName= DCIM_EnclosureTemperatureSensor+DeviceID=<FQDD>”

7.1.9.2 Operations

The following table lists the implemented operations on DCIM_EnclosureTemperatureSensor.

Table 19 – DCIM_EnclosureTemperatureSensor - Operations

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

7.1.9.3 Class Properties

The following table lists the implemented properties for DCIM_EnclosureTemperatureSensor instance representing the Enclosure Temperature Sensor in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 20 – DCIM_EnclosureTemperatureSensor - Properties

Property Name	Requirement	Type	Additional Requirements
SystemCreationClassName	Mandatory	String	The property value shall be “DCIM_ComputerSytem”.
SystemName	Mandatory	String	The property value shall be “DCIM:ComputerSystem”
CreationClassName	Mandatory	String	The property value shall be “DCIM_EnclosureTemperatureSensor”
DeviceID	Mandatory	String	The property shall have the sensor FQDD value.
FQDD	Mandatory	String	The property shall represent Fully Qualified Device Description (FQDD)
BaseUnits	Mandatory	uint16	The property value shall be 2 (Degrees C).
CurrentReading	Mandatory	sint32	The present value indicated by the sensor. The property value shall be in Degrees C.
ElementName	Mandatory	string	The property shall have the value “EnclosureTemperatureSensor”.
PrimaryStatus	Mandatory	uint32	The property shall represent the status of the device and shall be one of the

Property Name	Requirement	Type	Additional Requirements
			following values: • 0 - Unknown • 1 - OK • 2 - Degraded • 3 - Error
RateUnits	Mandatory	uint16	The property value shall be 0 (None).
SensorType	Mandatory	Uint16	The property value shall be 2 (Temperature).
UnitModifier	Mandatory	sint32	The property shall have the value 0 denoting that the CurrentReading property value need not be multiplied by the UnitModifier property value.
SettableThresholds[]	Mandatory	uint16	An array representing the writable thresholds supported by Sensor. The property shall be "null" value since this is read-only sensor
SupportedThresholds[]	Mandatory	uint16	The array property shall have following values: • 0 - LowerThresholdNonCritical • 1 - UpperThresholdNonCritical • 2 - LowerThresholdCritical • 3 - UpperThresholdCritical
UpperThresholdCritical	Mandatory	sint32	The property shall have upper critical threshold value in Degree Centigrade.
UpperThresholdNonCritical	Mandatory	sint32	The property shall have upper non-critical threshold value in Degree Centigrade.
LowerThresholdCritical	Mandatory	sint32	The property shall have lower critical threshold value in Degree Centigrade.
LowerThresholdNonCritical	Mandatory	sint32	The property shall have lower non-critical threshold value in Degree Centigrade.

7.2 Attributes

This section details the supported attributes for the storage devices. Not all attributes shall 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.

NOTE: 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: root/dcim.

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 lists the implemented operations on DCIM_RAIDEnumeration.

Table 11 – DCIM_RAIDEnumeration - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_RAIDService.SetAttributte()	Mandatory	See section 8.15.
DCIM_RAIDService.SetAttributes()	Mandatory	See section 8.16.

7.2.1.3 Class Properties

The following table lists the implemented properties for DCIM_RAIDEnumeration instance representing a RAID enumeration attribute. The "Additional Requirements" column shall denote either possible values for the property, or requirements on the value formulation.

560 **Table 12 – Class: DCIM_RAIDEnumeration**

Property Name	Requirement	Type	Additional Requirements
InstanceID	Mandatory	string	The property value shall be formed as follows: "<FQDD property value>:<AttributeName property value>".
AttributeName	Mandatory	string	The property value shall be from the "AttributeName" column in Table 13.
CurrentValue[]	Mandatory	string	The property value shall be one of the values in the "PossibleValues" column at the corresponding row in Table 13.
PendingValue[]	Mandatory	string	The property value shall be one of the values in the "PossibleValues" column at the corresponding row in Table 13.
IsReadOnly	Mandatory	boolean	The property value shall be from the "IsReadOnly" column in Table 13.
FQDD	Mandatory	string	FQDD of the device that the attribute belongs to.
PossibleValues[]	Mandatory	string	The property value shall be equal to the array of the values in "PossibleValues" column at the corresponding row in Table 13.

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

563 **Table 13 – DCIM_RAIDEnumeration Attributes**

AttributeName	Description	IsRead Only	PossibleValue
RAIDSupportedRAIDLevels	Supported RAID levels. This attribute relates to the controller device.	TRUE	RAID-0, RAID-1, RAID-5, RAID-10, RAID-50, RAID-60
RAIDsupportedDiskProt	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
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 polcy 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

AttributeName	Description	IsRead Only	PossibleValue
RAIDPDState	Physical Disk state. This attributes relates to the physical disk.	TRUE	Unknown, Ready, Online, Foreign, Blocked, Failed, Non-RAID, Missing
RAIDHotSpareStatus	Hotspare status. This attribute relates to the physical disk.	TRUE	No, Dedicated, Global
RAIDNegotiatedSpeed	NegotiatedSpeed. This attribute relates to the physical disk.	TRUE	1_5_GBS, 3_GBS, 6_GBS
RAIDSupportedInitTypes	Supported virtual disk initialization types. This attribute is related to virtual disk.	TRUE	Fast

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: root/dcim.

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 lists 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.SetAttributte()	Mandatory	See section 8.15.
DCIM_RAIDService.SetAttributes()	Mandatory	See section 8.16.

7.2.2.3 Class Properties

The following table lists the implemented properties for DCIM_RAIDString instance representing a RAID string attribute. The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 15 – Class: DCIM_RAIDString

Properties	Requirements	Type	Additional Requirements
InstanceID	Mandatory	string	The property value shall be formed as follows: "<FQDD property value>:<AttributeName property value>".
AttributeName	Mandatory	string	The property value shall be from the "AttributeName" column in Table 16.
CurrentValue[]	Mandatory	string	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	string	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	boolean	The property value shall be the value in the "R/RW" column at the corresponding row in Table 16.
FQDD	Mandatory	string	FQDD of the device that the attribute belongs to.
MinLength	Mandatory	uint64	The property value shall be the value in the "MinLength" column at the corresponding row in Table 16.
MaxLength	Mandatory	uint64	The property value shall be the value in the "MaxLength" column at the corresponding row in Table 16.

The following table lists 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
RAIDEffectiveSASAddress	EffectiveSASAddress. This attribute relates to enclosure.	TRUE	16	16

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: root/dcim.

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 lists 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.SetAttributte()	Mandatory	See section 8.15.
DCIM_RAIDService.SetAttributes()	Mandatory	See section 8.16.

7.2.3.3 Class Properties

The following table lists the implemented properties for DCIM_RAIDIntger instance representing a RAID integer attribute. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Property Name	Requirements	Type	Additional Requirements
InstanceID	Mandatory	string	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	string	The property value shall be from the “AttributeName” column in Table 18.
CurrentValue[]	Mandatory	string	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	string	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	Boolean	The property value shall be the value in the “IsReadOnly” column at the corresponding row in Table 18.
FQDD	Mandatory	string	FQDD of the device that the attribute belongs to.
LowerBound	Mandatory	uint64	The property value shall be the value in the “LowerBound” column at the corresponding row in Table 18.
UpperBound	Mandatory	uint64	The property value shall be the value in the “UpperBound” column at the corresponding row in Table 18.

Table 17 – Class: DCIM_RAIDInteger

The following table lists 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[1,2]	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
RAIDspinDownIdleTime	Spin down idle time of the controller. This attribute is related to the controller	TRUE	1	65535
RAIDNominalMediumRotationRate	Nominal medium rotation rate. This attribute is related to physical disk.	TRUE	2	4294967295

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

NOTE: 2 – The rebuild rate, configurable between 0% and 100%, represents the percentage of the system resources dedicated to rebuilding failed array disks. At 0%, the rebuild will have the lowest priority for the controller, will take the most time to complete, and will be the setting with the least impact to system performance. A rebuild rate of 0% does not mean that the rebuild is stopped or paused.

7.3 DCIM_RAIDService

This section describes the implementation for the DCIM_RAIDService class.

This class shall be instantiated in the Implementation Namespace: root/dcim.

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 de lists tails 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 Class Properties

The following table lists the implemented properties for DCIM_RAIDService instance representing a storage service in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 20 – Class: DCIM_RAIDService

Property Name	Requirement	Type	Additional Requirements
SystemCreationClassName	Mandatory	string	The property value shall be “DCIM_ComputerSystem”.
CreationClassName	Mandatory	string	The property value shall be “DCIM_RAIDService”.
SystemName	Mandatory	string	The property value shall be “DCIM:ComputerSystem”.
Name	Mandatory	string	The property value shall be “DCIM:RAIDService”.
ElementName	Mandatory	string	The property value shall be “RAID Service”.

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 lists 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 Class Properties

The following table lists the implemented properties for DCIM_LCRegisteredProfile instance representing RAID Profile implementation. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 22 – Class: DCIM_RegisteredProfile

Property Name	Requirement	Type	Description
InstanceID	Mandatory	String	DCIM:SimpleRAID:1.0.0
RegisteredName	Mandatory	String	This property shall have a value of “Simple RAID”.
RegisteredVersion	Mandatory	String	This property shall have a value of “1.3.0”.
RegisteredOrganization	Mandatory	Uint16	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	Mandatory	String	This property shall match “DCIM”
AdvertisedTypes[]	Mandatory	Uint16	This property array shall contain [1(Other), 1 (Other)].
AdvertiseTypeDescriptions[]	Mandatory	String	This property array shall contain ["WS-Identify", "Interop Namespace"].
ProfileRequireLicense[]	Mandatory	String	This property array shall describe the required licenses for this profile. If no license is required for the profile, the property shall have value NULL.

ProfileRequireLicenseStatus[]	Mandatory	String	This property array shall contain the status for the corresponding license in the same element index of the ProfileRequireLicense array property. Each array element shall contain: “LICENSED” “NOT_LICENSED” If no license is required for the profile, the property shall have value NULL.

8 Methods

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

8.1 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.
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

Qualifiers	Name	Type	Description/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

8.2 DCIM_RAIDService.ResetConfig()

The ResetConfig() method is used to delete all the virtual disks and unassign all hot spare physical disks.
CAUTION: 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.
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 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.
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 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 setting this virtual disk for deletion. The ObjectStatus and PendingOperations property in the Virtual Disk view has the value “PendingDelete”. The Virtual disk is not deleted until a configuration job is scheduled and the system is rebooted.

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

Value	Description
0	Request was successfully executed.
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 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 and unfinished creation of a virtual disk. The ObjectStatus and PendingOperations property in the Virtual Disk view class has the value “PendingCreate”. The virtual disk shall 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 virtual disk shall change.

This method also supports creation of sliced virtual disk. A sliced virtual disk shall be created if the Size input parameter value is less than total size of the physical disks. Additional sliced virtual disk may be created using the same set of physical disks and the same RAID level that was used to create the first virtual disk.

NOTE: If the set of physical disks already has sliced virtual disks, 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 described earlier. In this scenario, CreateVirtualDisk () method shall only accept the VDPropNameArray-VDPropValueArray pairs mentioned in following table.

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

VDPropNameArray values	***VDPropValueArray* Value Description**
Cachcade	The valid input value is 1. (required)
VirtualDiskName	Name (optional)

Table 36 – DCIM_RAIDService.CreateVirtualDisk() Method: VDProp

VDPropNameArray Name	**Requirement**	**Additional Requirements**
Size	Optional	Size (in MB) of the virtual disk.
RAIDLevel	Mandatory	The new RAID level such as 0, 1, 5, or 6
SpanDepth	Optional	Number of spans in virtual disk.
SpanLength	Mandatory only for multispan	Number of disks per span.
StripeSize	Optional	See DCIM_VirtualDiskView class (see section 7.1.3).
ReadPolicy	Optional	See DCIM_VirtualDiskView class (see section 7.1.3).
WritePolicy	Optional	See DCIM_VirtualDiskView class (see section 7.1.3).
DiskCachePolicy	Optional	See DCIM_VirtualDiskView class (see section 7.1.3).
VirtualDiskName	Optional	Name of the virtual disk..
Initialize		0 - Fast
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. For example, in hexadecimal format 0xFFFF. For example, in decimal format 65535.

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

Value	**Description**
0	Request was successfully executed.
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

MessageID (OUT parameter)	Message
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 specfied 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 DCIM_RAIDService.InitializeVirtualDisk()

The InitializeVirtualDisk() method is used to initialize a single virtual disk from the targeted controller.

The successful execution of this method results in setting this virtual disk for initialization. The PendingOperations property in the Virtual Disk view has the value “Fast Init”.

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

Value	Description
0	Request was successfully executed.
2	Error occurred

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

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR028	Virtual Disk not found
STOR061	Init mode not supported on RAID controller

Table 42 – DCIM_RAIDService.InitializeVirtualDisk () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	VirtualDisk	String	FQDD of the virtual disk to initialize
IN, REQ	InitType	uint16	The parameter shall have value 0 (Fast Init).
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

8.7 DCIM_RAIDService.GetDHSDisks ()

The GetDHSDisks() method is used to determine possible choices of physical drives that can used to set a dedicated hotspare 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 for hot spares.

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

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 44 – 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 45 – 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

8.8 DCIM_RAIDService.GetRAIDLevels()

The GetRAIDLevels() method is used to determine the possible choices of RAID Levels to create virtual disks. If the list of physical disks is not provided, this method accesses information for all the connected disks.

Table 46 – DCIM_RAIDService.GetRAIDLevels() Method: Return Code Values

Value	Description
0	Request was successfully executed.
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 47 – 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 Hard Disk only • 2 - Include Solid State Only
IN, REQ	Diskprotocol	Uinit32	• 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 or include only non-FDE disks • 2- Include only non-FDE disks
IN	PDArray[]	String	Array of FQDD(s) identifies the 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

8.9 DCIM_RAIDService.GetAvailableDisks ()

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

Table 48 –DCIM_RAIDService.GetAvailableDisks() Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 49 – 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 50 – 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 Hard Disk 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	
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

8.10 DCIM_RAIDService.CheckVDValues()

The CheckVDValues() method is used to determine the possible sizes of Virtual disks and the default settings, proviced a RAID level and set of physical disks.

The VDPropArray property is filled with Size and other values, so that the method is successfully executed. If the SpanDepth is not provided, a default value of 2 shall be used for RAID levels 10, 50, and 60. **NOTE:** For certain numbers of disks such as nine or fifteen, it may be necessary for the user to provide another SpanDepth.

Table 51 – DCIM_RAIDService.CheckVDValues() Method:

VDPropNameArrayIn Values	Requirment	Description
Size	Optional	Size (in MB) of the virtual disk.
SpanDepth	Optional	Number of spans in a virtual disk (required for multispan RAID level.) The default value is two for Multispan RAID levels and one for basic RAID levels
RAIDLevel	Mandatory	See RAIDLevel Values and ValueMaps from DCIM_VirtualDiskView MOF.

StartingLBA		Starting logical block address of virtual disks in 512 byte blocks. If input value is 0xFFFFFFFFFFFFFFFF or 18446744073709551615, startingLBA is calculated programmatically.

Table 52 – DCIM_RAIDService.CheckVDValues() Method:

VDPropNameArrayOut *values*	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	See RAIDLevel Values and ValueMaps from DCIM_VirtualDiskView MOF.
SpanDepth	Number of spans in virtual disk.
SpanLength	Number of disks per span.
StripeSize	See DCIM_VirtualDiskView class (see section 7.1.3).
ReadPolicy	See DCIM_VirtualDiskView class (see section 7.1.3).
WritePolicy	See DCIM_VirtualDiskView class (see section 7.1.3).
DiskCachePolicy	See DCIM_VirtualDiskView class (see section 7.1.3).
Name	Virtual disk name.
StartingLBA	Starting logical Block address in 512 byte blocks of the virtual disk.

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

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 54 –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

MessageID (OUT parameter)	Message
STOR035	Not enough Storage objects or Storage objects in incorrect state for this operation
STOR043	Physical Disk is part of Virtual Disk that is not Secondary Raid Level 0
STOR044	All Physical Disks specfied 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 55 – 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

8.11 DCIM_RAIDService.SetControllerKey()

The SetControllerKey() method is used to set the key on controllers and set the controller in Local key Management (LKM) to encrypt the drives.

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

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 57 –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 58 – 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 shall be maximum of 32 characters in length, where the expanded form of the special character is counted as a single character.. The Key shall have at least one character from each of the following sets. • Upper Case • Lower Case • Number • Special Character The special characters in the following set need to be passed as mentioned below. • & → &amp; • < → &lt; • > → &gt; • “ → &quot; • ‘ → &apos;
IN, REQ	Keyid	String	Key Identifier that describes the key. The Keyid shall be maximum of 32 characters in length and should not have any spaces.
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 DCIM_RAIDService.LockVirtualDisk ()

The LockVirtualDisk() method encrypts the virtual disk.

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

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 60 – 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 61 – 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

8.13 DCIM_RAIDService.CreateTargetedConfigJob()

The CreateTargetedConfigJob( ) method is used to apply the pending values set by different methods under DCIM_RAIDService class.

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

Value	Description
2	Error occurred
4096[1]	Job started: REF returned to started CIM_ConcreteJob[1]

Table 63 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	string	FQDD of target device (controller)

Qualifiers	Name	Type	Description/Values
IN	RebootJobType	uint16	Creates a specific reboot job to power cycle the host system. This parameter only creates the RebootJob and does not schedule it. Shall contain the requested reboot type: 1 - PowerCycle 2 - Graceful Reboot without forced shutdown 3 - Graceful Reboot with forced shutdown. NOTE: This parameter only creates the RebootJob and does not schedule it.
IN	ScheduledStartTime	string	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. Start time for the job execution in format: yyyymmddhhmmss. The string "TIME_NOW" means immediate.
IN	UntilTime	string	Defines a time window for scheduling the job(s). However, this parameter is dependent on “ScheduledStartTime” and “ScheduledStartTime” parameters. Once scheduled, jobs will be executed within the time window. End time for the job execution in format: yyyymmddhhmmss. : 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.[1]
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

NOTE: 1 – If return code is 4096 (Job Created), the newly created job will not execute if the LC core services are not running (DCIM_LCEnumeration with AttributeName equal to “LifecycleControllerState” has the CurrentValue property equal to “Disabled”).

NOTE: If CreateTargetedConfigJob method is executed without the 3 optional parameters discussed above, the 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”. For more information, see “Job Control Profile”.

Table 64 – 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
LC062	An instance of Export or Import System Configuration is already running.

8.14 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 65 – DCIM_RAIDService.DeletePendingConfiguration() Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 66 – 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
LC062	An instance of Export or Import System Configuration is already running.

Table 67 – DCIM_RAIDService.DeletePendingConfiguration () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)

Qualifiers	Name	Type	Description/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

8.15 DCIM_RAIDService.SetAttribute( )

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

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

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

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

Value	Description
0	Completed with no error
2	Error occurred

Implementation of standard messages is optional. Standard messages defined for this method are described in Table 69.

Table 69 – 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>
STOR042	Unsupported Attribute Value for Attribute Name <Attribute Name>
STOR047	AttributeValue cannot be changed for ReadOnly Attribute Name <Attribute Name>
LC062	An instance of Export or Import System Configuration is already running.

Table 70 – 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 shall already exist.
OUT	SetResult[]	String	Returns: • "Set CurrentValue" when the attribute's current value is set. • "Set PendingValue" when the attribute's pending value is set.
IN, REQ	AttributeValue[ ]	String	Shall contain a new value to assign to the specified attribute. If this value is valid, it is 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

8.16 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.

NOTE: 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.

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

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

Value	Description
0	Completed with no error
2	Error occurred

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

MessageID(OUT parameter)	Message
STOR006	General failure

MessageID(OUT parameter)	Message
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>
LC062	An instance of Export or Import System Configuration is already running.

Table 73 – DCIM_RAIDService.SetAttributes( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of of target device
IN, REQ	AttributeName[ ]	String	The array parameter shall contain the AttributeName property values for the attributes to be modified.
IN, REQ	AttributeValue[ ]	String	The array parameter shall contain the desired attribute values. If the value is valid, the CurrentValue or PendingValue property of the specified attribute will be modified.
OUT	SetResult[ ]	String	Returns: • ”Set CurrentValue" when the attribute’s current value is set. • “Set PendingValue” when the attribute’s pending value is set.
OUT	RebootRequired[]	String	Returns: • “Yes” if reboot is required. • “No” if reboot is not required.
OUT	MessageID[]	String	Error MessageID
OUT	Message[]	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

8.17 DCIM_RAIDService.RemoveControllerKey()

The RemoveControllerKey() method erases the encryption key on controller.

CAUTION: All encrypted drives shall be erased.

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

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 75 – 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 76 – 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.

8.18 DCIM_RAIDService.EnableControllerEncryption()

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

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

Value	Description
0	Request was successfully executed.
2	Error occurred

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

MessageID(OUT parameter)	Message
STOR0003	Missing parameter
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

MessageID(OUT parameter)	Message
STOR020	Controller Key is already present

Table 79 – 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 is the passcode. This parameter is required if the mode is set to Local Key Management. The Key shall be maximum of 32 characters in length, where the expanded form of the special character is counted as a single character. The Key shall have one character from each of the following set. • Upper Case • Lower Case • Number • Special Character. The special characters in the following set need to be passed as mentioned below. • & → &amp; • < → &lt; • > → &gt; • “ → &quot; • ‘ → &apos;
IN	Keyid	String	Key Identifier describes the Key. This parameter is required if the mode is set to Local Key Management. The Keyid shall be maximum of 32 characters in length and shoutd not have any spaces.
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 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 80 – DCIM_RAIDService. ReKey () Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 81 – 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 82 – 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)
IN	Newkey	String	New controller key. The key shall be maximum of 32 characters in length, where the expanded form of the special character is counted as a single character. The Key shall have one character from each of the following set. • Upper Case • Lower Case • Number • Special Character. The special characters in the following set need to be passed as mentioned below and are counted as a single character for the maximum length of the key. • & → &amp; • < → &lt; • > → &gt; • “ → &quot; • ‘ → &apos;
IN	Oldkey	String	Old controller key.
IN	Keyid	String	Key identifier describes the key. The Keyid shall be maximum 32 characters in length and shoutd not have any spaces.

Qualifiers	Name	Type	Description/Values
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

8.20 DCIM_RAIDService.UnassignSpare()

The UnassignSpare() 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 shall be available for use.

Table 83 – DCIM_RAIDService.UnassignSpare() Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 84 – DCIM_RAIDService.UnassignSpare() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR003	Missing 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

Table 85 – 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.

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

8.21 DCIM_RAIDService.ConvertToRAID()

The ConvertToRAID() method is used to convert a physical disks in Non-RAID state to a state usable for RAID. After the method is successfully executed the PendingValue property of RAIDPDState should reflect the pending changes. After the CreateTargetedConfigJob method is successfully executed the DCIM_PhysicalDiskView.RAIDStatus property of that physical disk should reflect the new state.

Table 83 – DCIM_RAIDService.ConvertToRAID() Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 84 – DCIM_RAIDService.ConvertToRAID() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure

Table 85 – DCIM_RAIDService.ConvertToRAID() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PDArray[]	String	This is an array of FQDDs of target devices (PhysicalDisk.)
OUT	RebootRequired	Uint8	This parameter shall indicate if reboot is required to set the value and shall have following values: • 0 – No • 1 – Yes
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

8.22 DCIM_RAIDService.ConvertToNonRAID()

The ConvertToNonRAID() method is used to convert a physical disks in RAID state of “Ready” to a Non-RAID state. After the method is successfully executed, the PendingValue property of RAIDPDState should reflect the pending changes. After the CreateTargetedConfigJob method is successfully executed, the DCIM_PhysicalDiskView.RAIDStatus property of that physical disk should reflect the new state.

Table 86 – DCIM_RAIDService.ConvertToNonRAID() Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 87 – DCIM_RAIDService.ConvertToNonRAID() Method: Standard Messages

MessageID(OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure

Table 88 – DCIM_RAIDService.ConvertToNonRAID() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PDArray[]	String	This is an array of FQDDs of target devices (PhysicalDisk)
OUT	RebootRequired	Uint8	This parameter shall indicate if reboot is required to set the value and shall be one of the following values: • 0 – No • 1 – Yes
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

See *Lifecycle Controller (LC) Integration Best Practices Guide*.

10 CIM Elements

No additional details specified.

11 Privilege and License Requirement

The following table lists the privilege and license requirements for the listed operations. For the detailed explanation of the privileges and licenses, refer to the Dell WSMAN Licenses and Privileges specification.

Table 89 – Privilege and License Requirements

Class and Method	Operation	User Privilege Required	License Required
DCIM_ControllerView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_EnclosureView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_PhysicalDiskView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY

Class and Method	Operation	User Privilege Required	License Required
DCIM_VirtualDiskView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_RAIDInteger	ENUMERATE, GET	Login	LM_REMOTE_CONFIGURATION
DCIM_RAIDString	ENUMERATE, GET	Login	LM_REMOTE_CONFIGURATION
DCIM_RAIDEnumeration	ENUMERATE, GET	Login	LM_REMOTE_CONFIGURATION
DCIM_ControllerBatteryView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_EnclosureEMMView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_EnclosureFanSensor	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_EnclosurePSUView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_EnclosureTemperatureSensor	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVENTORY
DCIM_RAIDService	ENUMERATE, GET	Login	NONE
DCIM_RAIDService. CreateTargetedConfigJob()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. DeletePendingConfiguration()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService.ResetConfig()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. CreateVirtualDisk()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. ClearForeignConfig()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. GetAvailableDisks()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. GetRAIDLevels()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_RAIDService.GetDHSDisks()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. CheckVDValues()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_RAIDService.AssignSpare()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService.InitializeVirtualDisk()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. DeleteVirtualDisk()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. LockVirtualDisk()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService. SetControllerKey()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService.SetAttribute()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RAIDService.SetAttributes()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RegisteredProfile. EnableControllerEncryption()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RegisteredProfile.	INVOKE	Login,	LM_REMOTE_CONFIGURATION

Class and Method	Operation	User Privilege Required	License Required
RemoveControllerKey()		System Control	
DCIM_RegisteredProfile.ReKey()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RegisteredProfile. UnassignSpare()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RegisteredProfile. ConvertToRAID()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_RegisteredProfile. ConvertToNonRAID()	INVOKE	Login, System Control	LM_REMOTE_CONFIGURATION
DCIM_LCRegisteredProfile	ENUMERATE, GET	Login	None.
DCIM_LCElementConformsToProfile	ENUMERATE, GET	Login	None.

ANNEX A
(informative)

Change Log

Version	Date	Description
1.3.0		Added the InitializeVirtualDisk() method Added RAIDSupportedInitTypes as DCIM_RAIDEnumeration attribute instance. Added the PendingOperations property to the DCIM_VirtualDiskView. Added LC062 error message to the SetAttribute(), SetAttributes(), CreateTargetedConfigJob(), and DeletePendingConfiguration() methods.