Simple RAID Profile

Document Number: DCIM1031
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2010-08-10

Version: 1.0.0

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

Simple RAID Profile

1 Scope

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

2 Normative References

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

2.1 Approved References

DMTF DSP0131, *Profile Registration Profile 1.0.0*

DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

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

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

2.2 Other References

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

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

3 Terms and Definitions

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

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

optionalcreate

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

3.4

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

3.5

`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

3.6

`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

4 Symbols and Abbreviated Terms

4.1

CIM

Common Information Model

FQDD

Fully Qualified Device Description

4.2

WBEM

Web-Based Enterprise Management

5 Synopsis

Profile Name: RAID

Version: 1.0.0

Organization: Dell

CIM Schema Version: 2.2

Central Class: DCIM_RAIDService

Scoping Class: CIM_ComputerSystem

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

Table 1 identifies profiles that are related to this profile.

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration Profile	DMTF	1.0	Mandatory

6 Description

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

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

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

The profile information is represented with the instance of CIM_RegisteredProfile.

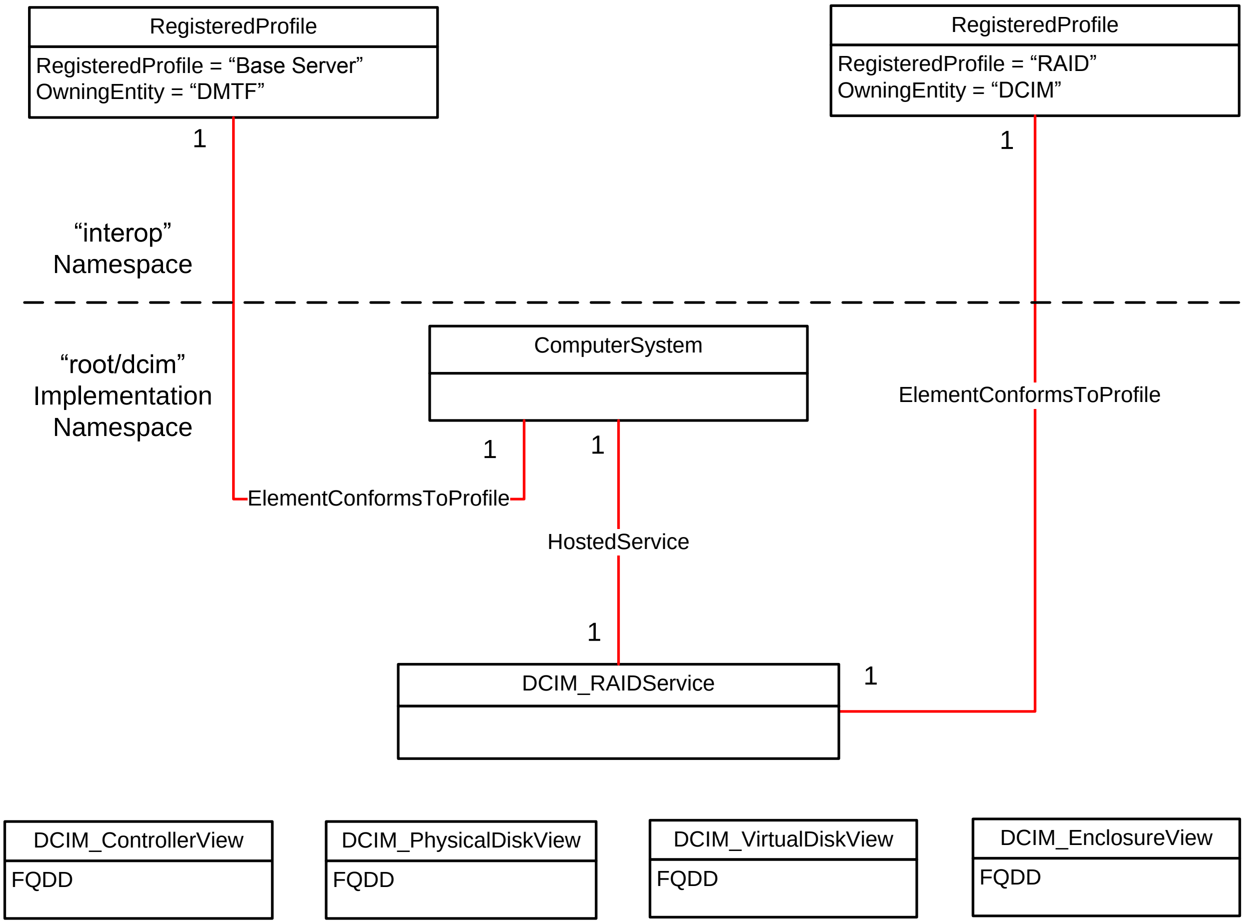

Figure 1 –RAID profile: Class Diagram

7 Implementation Requirements

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

A single instance of DCIM_RAIDService shall be instantiated and one instance of DCIM_ControllerView shall be instantiated. DCIM_RAIDService shall be associated with its view(s) through the FQDD property.

7.1 Views

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

7.1.1 Controller View

The following

Properties	Data type	Description
FQDD	string	Fully Qualified device description (uniquely identifies device)
InstanceID	string	The value is same as FQDD
Primary status	Uint32	Status of Device: Unknown, OK, Degraded, Error
Rollup Status	Uint32	Status of Children : Unknown, OK, Degraded, Error
Controller Firmware Version	string	The firmware version
PCI Slot number	Uint8	The associated PCI slot
Bus	string	PCI Bus
Device	string	PCI device
Function	string	PCI function
PCIVendorID	string	PCI vendor identifier
PCISubVendorID	string	PCI sub vendor identifier
PCIDeviceID	string	PCI device identifier
PCISubDeviceID	string	PCI sub device identifier
DeviceCardManufacturer	string	Manufacturer name
DeviceCardDataBusWidth	enum	Bus width: Unknown, 8x
DeviceCardSlotLength	enum	Slot length: Short, Long
DeviceCardSlotType	enum	Slot type: Unknown, PCI Express x8
SecurityStatus	enum	Controller security: Unknown, Encryption Capable, Security Key Assigned

Product Name	string	Marketing name of the controller
SAS Address	string	Hexadecimal- Provides unique id of controller

7.1.2 Enclosure View

Property	Data type	Description
FQDD	string	Fully Qualified device description (uniquely identifies device)
InstanceID	string	The value is same as FQDD
PrimaryStatus	Uint32	Status of Device: Unknown, OK, Degraded, Error
RollupStatus	Uint32	Status of children : Unknown, OK, Degraded, Error
Connector	Uint8	Controller port connection
WiredOrder	Uint8	Order from Controller port(0 for backplane)
ServiceTag	string	Up to 10 characters
AssetTag	string	Up to 10 characters
Version	string	Emm version
SlotCount	Uint8	Number of Drive slots
EMMCount	Uint8	Number Enclosure Management Modules present
PSUCount	Uint8	Number of Power supplies present
FanCount	Unit8	Number of fans present
TempProbeCount	Uint8	Number of temperature probes present
ProductName	string	Marketing name

7.1.3 Virtual Disk View

Properties	Data type	Description
FQDD	string	Fully Qualified device description(uniquely identifies device)
InstanceID	string	The value is same as FQDD
PrimaryStatus	Uint32	Status of Device: Unknown, OK, Degraded, Error
RAIDStatus	Uint32	RAID Specific status: Unknown, Ready, Online, Foreign, Offline, Blocked, Failed, Degraded
Write Cache Policy	Uint32	Current Policy : Write Through, Write Back, Write Back force

Read Cache Policy	Uint32	Current Policy : No Read Ahead, Read Ahead, Adaptive
RAID Types (Raid Level)	Uint32	RAID level: RAID-0 RAID-1 RAID-5 RAID-10 RAID-50 RAID-6 RAID-60
SizeInBytes	Uint64	The size of VD
Stripe Size	Uint32	The strip size: 512B, 1
Name	string	The Virtual Disk name
Span Length	Uint32	Physical disks per span
SpanDepth	Uint32	Number of spans in virtual disk
Physical Disk IDs	Array of string	Array of Physical disk FQDDs
Virtual Disk Target ID	Uint32	Virtual disk target number
RemainingRedundancy	Uint16	RemainingRedundancy
DiskCachePolicy	Uint32	Policy for physical disks included in the virtual disk: Default, Enabled, Disabled
ObjectStatus	Uint8	Virtual disk configuration state: Unknown PendingDelete PendingCreate Current

7.1.4 Physical Disk View

Properties	Data type	Description
FQDD	string	Fully Qualified device description(uniquely identifies device)
InstanceID	string	The value is same as FQDD
RAIDStatus	Uint32	RAID Specific status: Unknown, Ready, Online, Foreign, Offline, Blocked, Failed, Degraded
PrimaryStatus	Uint32	Status of device: Unknown, OK, Degraded, Error
Connector	Uint16	Controller Port.
Slot	Uint16	Slot where drive is located

SizeInBytes	Uint64	Coerced size of physical disk
Model	string	Model Name of the physical disk
Manufacturer	string	Manufacturer of the physical disk
ManufacturingDay	Uint16	Manufacturing Day
ManufacturingWeek	Uint16	Manufacturing Week
ManufacturingYear	Uint32	Manufacturing Year
Revision	string	revision number of Virtual disk
SerialNumber	string	Serial number of physical disk
BusProtocol	Uint32	Protocol: Unknown, SCSI, PATA, FIBRE, USB, SATA, SAS
HotSpareStatus	Uint16	Hotspare Status: No, Dedicated, Global
PredictiveFailureState	Uint32	Smart alert: Smart Alert Absent, Smart Alert Present
SecurityState	Uint32	Security State: Unknown, Secured, Locked, Foreign
MediaType	Uint32	Drive media type: Magnetic Drive, Solid State Drive
FreeSizeInBytes	Uint64	Free space available for use by a virtual disk
UsedSizeInBytes	Uint64	Space already Consumed by Virtual Disks
MaxCapableSpeed	Uint32	Speed disk capable of: Unknown, 1.5GBS, 3GBS, 6GBS
SASAddress	string	SAS address of drive

8 Methods

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

8.1 Method: DCIM_RAIDService.AssignSpare ()

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

Table 2 – DCIM_RAIDService.AssignSpare ( ) Method: Return Code Values

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

Table 3 – 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 4 – DCIM_RAIDService.AssignSpare () Method: Parameters

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

8.2 Method: DCIM_RAIDService.ResetConfig ()

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

Table 5 – DCIM_RAIDService.ResetConfig () Method: Return Code Values

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

Table 6 – 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 7 – 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 (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.3 Method: DCIM_RAIDService.ClearForeignConfig ()

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

Table 8 – DCIM_RAIDService.ClearForeignConfig ( ) Method: Return Code Values

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

Table 9 – 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 10 – 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 (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.4 Method: DCIM_RAIDService.DeleteVirtualDisk ()

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

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

Table 11 – DCIM_RAIDService.DeleteVirtualDisk() Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter

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

Table 12 – 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 (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.5 Method: DCIM_RAIDService.CreateVirtualDisk ()

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

Table 13 – DCIM_RAIDService.CreateVirtualDisk () Method: VDProp

VDPropNameArray values	*VDPropValueArray* Value Description
Size	Size, in MB, of the virtual disk.(optional)
RAIDLevel	The new RAID level, such as 0, 1, 5, or 6..(required)
SpanDepth	Number of spans in virtual disk.(required for multispan)
SpanLength	Number of disks per span.(required for multispan)
StripeSize	8k, 16k, 32k … (optional)
ReadPolicy	(optional)
WritePolicy	(optional)
DiskCachePolicy	(optional)
VirtualDiskName	Name (optional)
Initialize	0 –Fast (Only option)

Table 14 – DCIM_RAIDService.CreateVirtualDisk () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred
3	Warning Optional parameter not applied

Table 15 – DCIM_RAIDService.CreateVirtualDisk () Method: Standard Messages

MessageID (OUT parameter)	Message
STOR003	Missing parameter
STOR004	Invalid parameter value
STOR006	General failure
STOR007	Resource Allocation Failure
STOR010	RAID level not supported on controller
STOR011	Stripe size not supported on controller
STOR012	Provided Physical disk configuration not valid
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

Table 16 – DCIM_RAIDService.CreateVirtualDisk () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (controller)
IN, REQ	PDArray	String Array	Array of FQDDs where each FQDD identifies a physical disk..
IN, REQ	VDPropNameArray	String Array	Indexed array of Virtual Disk property names with relative values contained in VDPropValueArray parameter.
IN, REQ	VDPropValueArray	String Array	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 (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.

Qualifiers	Name	Type	Description/Values
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.6 Method: DCIM_RAIDService.GetDHSDisks ()

The GetDHSDisks () method is used to determine possible choices of drives to be 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 hot spares.

Table 17 – DCIM_RAIDService.GetDHSDisks () Method: Return Code Values

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

Table 18 – 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 19 – DCIM_RAIDService.GetDHSDisks () Method: Parameters

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

8.7 Method: DCIM_RAIDService.GetRAIDLevels ()

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

Table 20 – DCIM_RAIDService.GetRAIDLevels () Method: Return Code Values

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

Table 21 – 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 22 – DCIM_RAIDService.GetRAIDLevels () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN, REQ	DiskType	Uinit32	0- Include all Types, 1- Include Magnetic only, 2 Include Solid State Only
IN, REQ	Diskprotocol	Uinit32	0 - Include all protocols, 1- Include Sata, 2 Include SAS
IN (optional)	DiskEncrypt	Uint32	0 – Include FDE (encryption capable) and Non encryption capable disks 1 – Include FDE only, include only non FDE disks
IN	PDArray	String Array	Array of FQDD(s) identifies physical disk(s)..
OUT	VDRAIDEnumArray	String Array	Indexed array of Virtual Disk RAID level enum values .
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

[Implementation Note: Supported methods include the following:

8.8 Method: DCIM_RAIDService.GetAvailableDisks ()

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

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

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

Table 24 – 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 25 – DCIM_RAIDService.GetAvailableDisks () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN, REQ	DiskType	Uint32	0 - Include all Types, 1- Include Magnetic only, 2 Include Solid State Only
IN, REQ	Diskprotocol	Uint32	0 - Include all protocols, 1- Include Sata, 2 Include SAS
IN	DiskEncrypt	Uint32	0 – Include FDE (encryption capable) and Non encryption capable disks 1 – Include FDE only, include only non FDE disks
IN	RaidLevel	Uint32	
OUT	PDArray	String Array	Array of FQDD(s) identifies physical disk(s)..
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.9 Method: DCIM_RAIDService.CheckVDValues ()

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

Table 26 – DCIM_RAIDService.CheckVDValues() Method:

VDPropNameArrayIn values	***VDPropValueArrayIn*** **Value Description**
Size	Size, in MB, of the virtual disk.(optional)
SpanDepth	Number of spans in virtual disk (optional)
RAIDLevel	The new RAID level, such as 0, 1, 5, or 6..(required)

Table 27 – DCIM_RAIDService.CheckVDValues() Method:

VDPropNameArrayOut values	***VDPropValueArrayOut*** **Value Description**
Size	Size, in MB, of the virtual disk.
RAIDLevel	The new RAID level, such as 0, 1, 5, or 6..
SpanDepth	Number of spans in virtual disk.
SpanLength	Number of disks per span.
StripeSize	Refer to Virtual Disk view
ReadPolicy	Refer to Virtual Disk view
WritePolicy	Refer to Virtual Disk view
DiskCachePolicy	Refer to Virtual Disk view
VirtualDiskName	Name

]

Table 28 -- DCIM_RAIDService.CheckVDValues () Method: Return Code Values

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

Table 29 – 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
STOR012	Provided Physical disk configuration not valid

Table 30 – DCIM_RAIDService.CheckVDValues () Method: Parameters

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

8.10 Method: DCIM_RAIDService.SetControllerKey ()

The SetControllerKey() method sets the key on controllers that support encryption of the drives.

Table 31 – DCIM_RAIDService.SetControllerKey () Method: Return Code Values

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

Table 32 – 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

Table 33 – 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
IN, REQ	Keyid	String	Key Identifier- Describes Key
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 (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.11 Method: DCIM_RAIDService.LockVirtualDisk ()

The LockVirtualDisk() method encrypts the virtual disk.

Table 34 – DCIM_RAIDService.LockVirtualDisk () Method: Return Code Values

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

Table 35 – 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 36 – 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

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.12 Method: DCIM_RAIDService.CreateTargetedConfigJob()

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

CreateTargetedConfigJob method supports the following optional input parameters

1. RebootJobType: when provided in the input parameters, creates a specific reboot job to "PowerCycle" or "Graceful Reboot without forced shutdown" or "Graceful Reboot with forced shutdown". This parameter only creates the RebootJob and does not schedule it.

2. ScheduledStartTime: When provided in the input parameters, schedules the "configuration job" and the optional "reboot job" at the specified start time. A special value of "TIME_NOW" schedules the job(s) immediately.

3. UntilTime: This parameter has a dependency on "ScheduledStartTime", together "ScheduledStartTime" and "UntilTime" define a time window for scheduling the job(s). Once scheduled, jobs will be executed within the time window.

If CreateTargetedConfigJob method is executed without the 3 optional parameters discussed above, then configuration job is created but not scheduled. However, this configuration job can be scheduled later using the DCIM_JobService.SetupJobQueue () method from the "Job Control Profile". DCIM_JobService.SetupJobQueue () can be executed to schedule several configuration jobs including the reboot job. Refer to "Job Control Profile" for more details.

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

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

Table 38 – 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

MessageID (OUT parameter)	Message
STOR023	Configuration already committed, cannot set configuration

Table 39 – DCIM_RAIDService.CreateTargetedConfigJob() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
IN	RebootJobType	uint16	Shall contain the requested reboot type: 1 - PowerCycle 2 - Graceful Reboot without forced shutdown 3 - Graceful Reboot with forced shutdown.
IN	ScheduledStartTime	string	Start time for the job execution in format: yyyymmddhhmmss. The string "TIME_NOW" means immediate.
IN	UntilTime	string	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.
OUT	MessageID	string	Error Message ID- can be used to index into Dell Message registry files
OUT	Message	string	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments	string[]	Substitution variables for dynamic error messages

8.13 Method: DCIM_RAIDService.DeletePendingConfiguration ()

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

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

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

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

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

Table 42 – DCIM_RAIDService.DeletePendingConfiguration () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of target device (Controller)
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.14 Profile Conventions for Operations

Support for operations for each profile class (including associations) is specified in the following subclauses. Each subclause includes either the statement "All operations in the default list in section 8.14 are supported as described by DSP0200 version 1.2" or a table listing all of the operations that are not supported by this profile or where the profile requires behavior other than that described by DSP0200.

The default list of operations is as follows:

- GetInstance
- EnumerateInstances
- EnumerateInstanceNames
- Associators
- AssociatorNames
- References
- ReferenceNames

A compliant implementation shall support all of the operations in the default list for each class, unless the "Requirement" column states something other than *Mandatory*.

9 Use Cases

This section contains use cases for the Dell RAID Profile.

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

9.1 Discovery of RAID profile support

Use one of the two procedures below to confirm the existence of RAID profile support

A) `GET` the *DCIM_LCRegisteredProfile* instance using an *InstanceID* of DCIM:SimpleRAID:1.0.0. See section 3.6 for a definition of `GET`.

Instance URI:

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

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

```
DCIM_LCRegisteredProfile
  AdvertiseTypeDescriptions = WS-Identify, Interop Namespace
  AdvertiseTypes = 1, 1
  InstanceID = DCIM:SimpleRAID:1.0.0
  OtherRegisteredOrganization = DCIM
  RegisteredName = Simple RAID
  RegisteredOrganization = 1
```

RegisteredVersion = 1.0.0

B) ENUMERATE the *CIM_RegisteredProfile* class. See section 3.5 for a definition of ENUMERATE.

Class URI:

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

Then query the result for the following properties:

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

9.2 Inventory of RAID controllers in system

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

Class URI:

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

The instance information of all available RAID controllers will be returned

9.3 Get the first RAID controller's information

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

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

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

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

9.4 Inventory of virtual disks in system

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

Class URI:

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

The instance information of all available virtual disks will be returned

9.5 Inventory of physical disks in system

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

Class URI:

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

The instance information of all available physical disks will be returned

9.6 Apply pending values for a particular RAID configuration

A) To invoke the CreateTargetedConfigJob() method, construct input parameters per Table 39 and use the RAID FQDD from section **Error! Reference source not found.**

B) INVOKE CreateTargetedConfigJob() method

Class URI:

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

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

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

9.7 Delete pending values for a particular RAID configuration

A) To invoke the DeletePendingConfiguration() method, construct input parameters per Table 42 and use the RAID FQDD from section **Error! Reference source not found.**

B) INVOKE DeletePendingConfiguration() method

Class URI:

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

C) If the return parameters indicate success**,** per Table 40, no further action necessary

9.8 Clear old configuration from newly added hard drives

A) To invoke the ClearForeignConfig() method, construct input parameters per Table 10 and use the particular RAID's FQDD

B) INVOKE ClearForeignConfig() method

Class URI:

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

C) If return message indicates success per Table 8, no further action necessary

9.9 Determine available RAID configurations for a given set of disks

A) To invoke the GetRAIDLevels( ) method, construct input parameters per Table 22 and use the desired RAID FQDDs from Section **Error! Reference source not found.**

B) INVOKE GetRAIDLevels( ) method

Class URI:

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

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

9.10 Determine available physical disks for a given RAID configuration

A) To invoke the GetAvailableDisks( ) method, construct input parameters per Table 25 and use the desired RAID FQDDs from Section **Error! Reference source not found.**

B) INVOKE GetAvailableDisks( ) method

Class URI:

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

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

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

A) To invoke the CheckVDValues() method, construct input parameters per Table 30 and using physical disk FQDDs from Section **Error! Reference source not found.** and a RAID level rom Section **Error! Reference source not found.**

B) INVOKE CheckVDValues() method

Class URI:

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

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

9.12 Create a virtual disk on the system

A) Determine applicable RAID configuration from Section **Error! Reference source not found.**

B) Select physical disks for RAID configuration from Section **Error! Reference source not found.**

C) Check available sizes and default virtual disk parameters in Section **Error! Reference source not found.**

D) To invoke the createVirtualDisk() method, construct input parameters per Table 16 and use the particular RAID FQDDs from Section **Error! Reference source not found.**

E) `INVOKE` createVirtualDisk() method

Class URI:

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

F) Examine output parameters per Table 14

G) Apply the pending values (Section **Error! Reference source not found.**) using the FQDD btained from Section **Error! Reference source not found.**

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

I) List virtual disk inventory, Section **Error! Reference source not found.**, to check the status f the creation of the new virtual disk

9.13 Determine available physical disks to be used as a hotspare

A) To invoke the GetDHSDisks() method, construct input parameters per Table 19 and use the applicable RAID virtual disk FQDD from Section **Error! Reference source not found.**

B) `INVOKE` GetDHSDisks() method

Class URI:

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

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

9.14 Assign a physical disk as a hotspare

A) Confirm at least one virtual disk exists, see Section **Error! Reference source not found.**

B) Determine potential physical disks to be used as a hotspare from Section **Error! Reference ource not found.**

C) To invoke the AssignSpare() method, construct input parameters per Table 4 and use the particular RAID virtual disk FQDD from Section **Error! Reference source not found.**

D) `INVOKE` AssignSpare() method

Class URI:

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

E) Examine output parameters per Table 2

F) Apply the pending values (Section **Error! Reference source not found.**) using the FQDD btained from Section **Error! Reference source not found.**

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

H) Confirm the applicable physical disk is a hotspare by listing the physical disks, per Section **Error! Reference source not found.**, and checking that the *HotSpareStatus* attribute is set o '1'

9.15 Delete a virtual disk from the system

A) To invoke the DeleteVirtualDisk( ) method, construct input parameters per Table 12 and use the particular RAID's FQDD from Section **Error! Reference source not found.**

B) `INVOKE` DeleteVirtualDisk ( ) method

Class URI:

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

C) Examine output parameters per Table 11

D) Apply the pending values (Section **Error! Reference source not found.**) using the FQDD btained from Section **Error! Reference source not found.**

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

F) Confirm successful execution of the method by listing the virtual disks, per Section **Error! eference source not found.**. The deleted virtual disk should not be displayed

9.16 Delete all virtual disks and unassign all hotspares

A) To invoke the ResetConfig( ) method, construct input parameters per Table 7 and use the FQDD from Section **Error! Reference source not found.**

B) `INVOKE` ResetConfig( ) method

Class URI:

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

C) Examine output parameters per Table 5

D) Apply the pending values (Section **Error! Reference source not found.**) using the FQDD btained from Section **Error! Reference source not found.**

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

F) Confirm successful execution of the method by listing the virtual disks and physical disks, per Section **Error! Reference source not found.** and Section **Error! Reference source not und.**, respectively. No virtual disks should be displayed and all physical disk *HotSpareStatus* attributes should be set to '0'

9.17 Encrypt a virtual disk

A) To invoke the SetControllerKey() method, construct input parameters per Table 33 and use the FQDD from Section **Error! Reference source not found.**

B) INVOKE SetControllerKey() method

Class URI:

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

C) To invoke the LockVirtualDisk() method, construct input parameters per Table 36 and use the virtual disk FQDD from Section **Error! Reference source not found.**

D) INVOKE LockVirtualDisk() method

Class URI:

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

E) Apply the pending values (Section **Error! Reference source not found.**) using the FQDD btained from Section **Error! Reference source not found.**

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

10 CIM Elements

Table 43 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 43. Sections 7.1 ("Implementation Requirements" and "Methods") may impose additional requirements on these elements.

Table 43 – CIM Elements: RAID ProfileRAID profile

Element Name	Requirement	Description
Classes		
DCIM_RAIDService	Mandatory	See section 10.1
DCIM_ControllerView	Mandatory	See section 10.2
DCIM_EnclosureView	Mandatory	See section 10.3
DCIM_VirtualDiskView	Mandatory	See section 10.4
DCIM_PhysicalDiskView	Mandatory	See section 10.5
DCIM_LCElementConformsToProfile	Mandatory	See section 10.6
DCIM_LCRegisteredProfile	Mandatory	See section 10.6
Indications		
None defined in this profile		

10.1 DCIM_RAIDService

DCIM_RAIDService is used to represent RAID.

Table 44 – Class: DCIM_RAIDService

Properties and Methods	Requirement	Description
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
ElementName	Mandatory	This property shall have value of "DCIM RAID Service".
AssignSpare ()	Mandatory	See section 8.1
ResetConfig ()	Mandatory	See section 8.2
ClearForeignConfig ()	Mandatory	See section 8.3
DeleteVirtualDisk ()	Mandatory	See section 8.4
CreateVirtualDisk ()	Mandatory	See section 8.5
GetDHSDisks ()	Mandatory	See section 8.6
GetRAIDLevels ()	Mandatory	See section 8.7
GetAvailableDisks ()	Mandatory	See section 8.8
CheckVDValues ()	Mandatory	See section 8.9
SetControllerKey ()	Mandatory	See section 8.10
LockVirtualDisk ()	Mandatory	See section 8.11
CreateTargetedConfigJob ()	Mandatory	See section 8.12

10.2 DCIM_ControllerView

The DCIM_ControllerView class groups together a set of Controller properties.

Table 45 – Class: DCIM_ControllerView

Properties	Notes	Description
FQDD	Mandatory	Fully Qualified device description(uniquely identifies device)
PrimaryStatus	Mandatory	Status of Device: Unknown, OK, Degraded, Error
RollupStatus	Mandatory	Status of Children : Unknown, OK, Degraded, Error
ControllerFirmwareVersion	Mandatory	The firmware version
PCISlot	Mandatory	The associated PCI slot
Bus	Mandatory	PCI Bus
Device	Mandatory	PCI device
Function	Mandatory	PCI function
PCIVendorID	Mandatory	PCI vendor identifier
PCISubVendorID	Mandatory	PCI sub vendor identifier
PCIDeviceID	Mandatory	PCI device identifier
PCISubDeviceID	Mandatory	PCI sub device identifier
DeviceCardManufacturer	Mandatory	Manufacturer name
DeviceCardDataBusWidth	Mandatory	Bus width: Unknown, 8x
DeviceCardSlotLength	Mandatory	Slot length: Short, Long
DeviceCardSlotType	Mandatory	Slot type: Unknown, PCI Express x8
SecurityStatus	Optional	Controller security: Unknown, Encryption Capable, Security Key Assigned
ProductName	Mandatory	Marketing name of the controller
SASAddress	Mandatory	Provides unique id of controller

10.3 DCIM_EnclosureView

The DCIM_EnclosureView class groups together a set of Enclosure properties.

Table 46 – Class: DCIM_EnclosureView

Properties	Notes	Description
FQDD	Mandatory	Fully Qualified device description(uniquely identifies device)
PrimaryStatus	Mandatory	Status of Device: Unknown, OK, Degraded, Error

RollupStatus	Mandatory	Status of children : Unknown, OK, Degraded, Error
Connector	Mandatory	Controller port connection
WiredOrder	Mandatory	Order from Controller port(0 for backplane)
ServiceTag	Optional	Up to 10 characters
AssetTag	Optional	Up to 10 characters
Version	Optional	Emm version
SlotCount	Mandatory	Number of Drive slots
EMMCount	Optional	Number Enclosure Management Modules present
PSUCount	Optional	Number of Power supplies present
FanCount	Optional	Number of fans present
TempProbeCount	Optional	Number of temperature probes present
ProductName	Mandatory	Marketing name

10.4 DCIM_VirtualDiskView

The DCIM_VirtualDiskView class groups together a set of Virtual disk properties

Table 47 – Class: DCIM_VirtualDiskView

Properties	Notes	Description
FQDD	Mandatory	Fully Qualified device description(uniquely identifies device)
PrimaryStatus	Mandatory	Status of Device: Unknown, OK, Degraded, Error
RAIDStatus	Mandatory	RAID Specific status: Unknown, Ready, Online, Foreign, Offline, Blocked, Failed, Degraded
Write Cache Policy	Optional	Current Policy : Write Through, Write Back, Write Back force
Read Cache Policy	Optional	Current Policy : No Read Ahead, Read Ahead, Adaptive
RAID Types (Raid Level)	Mandatory	RAID level: RAID-0 RAID-1 RAID-5 RAID-10 RAID-50 RAID-6 RAID-60

Size	Mandatory	The size of VD
Stripe Size	Mandatory	The strip size
Name	Optional	The Virtual Disk name
Span Length	Mandatory	Physical disks per span
Numberof Spans	Mandatory	Number of spans in virtual disk
Physical Disk IDs	Mandatory	Array of Physical disk FQDDs
Virtual Disk Target ID	Mandatory	Virtual disk target number
RemainingRedundancy	Mandatory	RemainingRedundancy
DiskCachePolicy	Mandatory	Policy for physical disks included in the virtual disk: Default, Enabled, Disabled
ObjectStatus	Mandatory	Virtual disk configuration state: Unknown PendingDelete PendingCreate Current

10.5 DCIM_PhysicalDiskView

The DCIM_PhysicalDiskView class groups together a set of Physical disk properties.

Table 48 – Class: DCIM_PhysicalDiskView

Properties	Notes	Description
FQDD	Mandatory	Fully Qualified device description(uniquely identifies device)
RAIDStatus	Mandatory	RAID Specific status: Unknown, Ready, Online, Foreign, Offline, Blocked, Failed, Degraded
PrimaryStatus	Mandatory	Status of device: Unknown, OK, Degraded, Error
Connector	Mandatory	Controller Port.
Slot	Mandatory	Slot where drive is located
Size	Mandatory	Coerced size of physical disk
Model	Optional	Model Name of the physical disk
Manufacturer	Optional	Manufacturer of the physical disk
ManufacturingDay	Optional	Manufacturing Day
ManufacturingWeek	Optional	Manufacturing Week
ManufacturingYear	Optional	Manufacturing Year
Revision	Optional	revision number of physical disk
SerialNumber	Optional	Serial number of physical disk

BusProtocol	Mandatory	Protocol: Unknown, SCSI, PATA, FIBRE, USB, SATA, SAS
HotSpareStatus	Mandatory	Hotspare Status: No, Dedicated, Global
PredictiveFailureState	Mandatory	Smart alert: Smart Alert Absent, Smart Alert Present
SecurityState	Optional	Security State: Unknown, Secured, Locked, Foreign
MediaType	Mandatory	Drive media type: Magnetic Drive, Solid State Drive
FreeSize	Mandatory	Free space available for use by a virtual disk
UsedSize	Mandatory	Space already Consumed by Virtual Disks
MaxCapableSpeed	Mandatory	Speed disk capable of: Unknown, 1.5GBS, 3GBS, 6GBS
SASAddress	Optional	SAS address of drive

10.6 DCIM_LCRegisteredProfile

The CIM_RegisteredProfile class (parent of DCIM_LCRegisteredProfile) is defined by the Profile Registration Profile. The requirements denoted in Table 49 are in addition to those mandated by the Profile Registration Profile.

The DCIM_LCRegisteredProfile shall be referenced by DCIM_LCElementConformsToProfile association.

Table 49 – Class: CIM_RegisteredProfile

Properties	**Requirement**	**Description**
RegisteredName	Mandatory	This property shall have a value of “DCIM RAID”
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”
RegisteredOrganization	Mandatory	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	Mandatory	This property shall match “Dell”

ANNEX A
(informative)

Related MOF Files

Dell Tech Center MOF Library:

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

Related Managed Object Format (MOF) files:

DCIM_ControllerView.mof

DCIM_EnclosureView.mof

DCIM_PhysicalDiskView.mof

DCIM_RAIDService.mof

DCIM_VirtualDiskView.mof

DCIM_LCElementConformsToProfile.mof

DCIM_LCRegisteredProfile.mof