Persistent Storage Profile

CONTENTS

Figures

Tables

Persistent Storage Profile

1   Scope

The Persistent Storage Profile extends the management capabilities of referencing profiles by adding the capability to represent and manage the partitions on the Virtual flash media on Dell platforms. The information in this specification is intended to be sufficient to manage the Persistent Storage feature using the DMTF CIM Schema and Dell extensions to the CIM schema.

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 DSP1033, *Profile Registration Profile 1.0.0*

DMTF DSP1061, *Management 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*

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

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

- DCIM_PersistentStorageService.mof

- DCIM_VFlashView

- DCIM_OpaqueManagementData

- DCIM_LCElementConformsToProfile

- DCIM_LCRegisteredProfile

3   Terms and Definitions

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

3.1
can

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

3.2
cannot

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

3.3
conditional

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

3.4
mandatory

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

3.5
may

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

3.6
need not

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

3.7
optional

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

3.8
referencing profile

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

3.9
shall

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

3.10
shall not

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

3.11
should

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

3.12
should not

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

---

3.13
`ENUMERATE`

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

3.14
`GET`

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

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 Manager Controller – management controller for the modular chassis

4.4
iSCSI

Internet Small Computer System Interface, an Internet Protocol (IP)-based storage networking standard for linking data storage facilities.

4.5
SD Card

Secure Digital Card.

4.6
AMEA

Advanced Management Enablement Adapter.

4.7
FQDD

Fully Qualified Device Description.

4.8
vFlash

Virtual Flash.

4.9
WBEM
Web-Based Enterprise Management

5   Synopsis

Profile Name: Persistent Storage

Version: 1.0.0

Organization: Dell Inc.

CIM Schema Version: 2.26 Experimental

Central Class: DCIM_PersistentStorageService

Scoping Class: CIM_ComputerSystem

The Persistent Storage Profile extends the management capability of the referencing profiles by adding the capability to represent and manage the partitions on the virtual flash media of Dell platforms. The Scoping Instance shall be the instance of CIM_System with which the Central Instance of DCIM_PersistentStorageService is associated through CIM_HostedService.

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
BIOS and Boot Management Profile	DCIM	1.0	Reference

6   Description

The Persistent Storage Profile describes the necessary properties and methods for representing and managing the partitions on the virtual flash media(SD Card on AMEA) provided by the iDRAC in Dell platforms.

The partition management of the virtual flash media includes:

- Listing virtual flash partitions
- Creating new partitions
- Deleting existing partitions
- Formatting a partition
- Exposing the partition in the host OS
- Detaching an attached partition
- Uploading an image to a partition
- Booting to a partition – see BIOS and Boot Management Profile
- Modifying a partition
- Exporting the contents of the partition

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

The vFlash media and its attributes are represented by the DCIM_VFlashView class. Each partition on the vFlash is represented by DCIM_OpaqueManagementData.

The DCIM_PersistentStorageService class is used to configure the vFlash and its partitions.

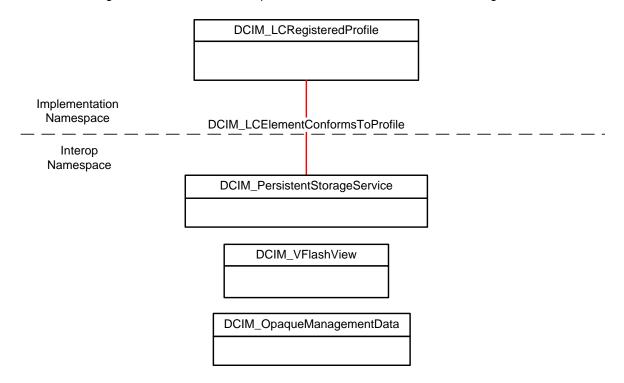The Persistent Storage Profile information is represented with the instance of CIM_RegisteredProfile.

Figure 1 – Persistent Storage Profile: Class Diagram

7   Implementation Requirements

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

Table 2 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 2. Sections 7 ("Implementation Requirements" and "Methods") may impose additional requirements on these elements.

Table 2 – CIM Elements: Persistent Storage Profile

Element Name	Requirement	Description
Classes		
DCIM_PersistentStorageService	Mandatory	The class maybe implemented in the Implementation Namespace. See sections 7.3
DCIM_VFlashView	Mandatory	The class shall be implemented in the Implementation Namespace. See section 7.1
DCIM_OpaqueManagementData	Mandatory	The class shall be implemented in the Implementation Namespace. See section 7.2
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in the *Implementation Namespace.*
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in the *Interop Namespace.*
DCIM_LCRegisteredProfile	Mandatory	The class shall be implemented in the Interop Namespace. See section 7.4
Indications		
None defined in this profile		

7.1 vFlash View

This section describes the implementation for the DCIM_VFlashView class.

This class shall be instantiated in the Implementation Namespace.

7.1.1 Resource URIs for WinRM®

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

The key property shall be the InstanceID.

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

7.1.2 Operations

The following table details the implemented operations on DCIM_VFlashView.

Table 3 – DCIM_VFlashView - Operations

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

Operation Name	Requirements	Required Input
DCIM_PersistentStorageService.InitializeMedia()	Mandatory	See section 8.1
DCIM_PersistentStorageService.VFlashStateChange()	Mandatory	See section 8.2

7.1.3   Properties

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

Table 4 – DCIM_VFlashView - Operations

Property Name	Requirement	Type	Additional Requirements
InstanceID	Mandatory	string	The property value shall be the FQDD property value.
FQDD	Mandatory	string	A string containing the Fully Qualified Device Description a user-friendly name for the object.
AvailableSize	Mandatory	uint64	The property value shall be in MB. This property specifies the available size on the media.
Capacity	Mandatory	uint64	The property value shall be in MB. This property specifies the total size on the media.
ComponentName	Mandatory	string	This property represents the media.
HealthStatus	Mandatory	string	The HealthStatus property represents the health status of the virtual flash media.
InitializedState	Mandatory	string	The InitializedState property represents the initialization state of the virtual flash media.
Licensed	Mandatory	boolean	This property represents whether the virtue flash media is licensed or not.
VFlashEnabledState	Mandatory	boolean	The EnabledState property indicates whether VFlash is enabled.
WriteProtected	Mandatory	boolean	This property indicates whether the virtue flash media is write protected (latch is on) or not.
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

7.2   DCIM_OpaqueManagementData

This section describes the implementation for the DCIM_OpaqueManagementData class.

This class shall be instantiated in the Implementation Namespace.

7.2.1   Resource URIs for WinRM®

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

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

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

7.2.2   Operations

The following table details the implemented operations on DCIM_OpaqueManagementData.

Table 5 – DCIM_OpaqueManagementData - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_PersistentStorageService.InitializeMedia()	Mandatory	See section 8.1
DCIM_PersistentStorageService.VFlashStateChange()	Mandatory	See section 8.2
DCIM_PersistentStorageService.CreatePartition()	Mandatory	See section 8.3
DCIM_PersistentStorageService.CreatePartitionUsingImage()	Mandatory	See section 8.4
DCIM_PersistentStorageService.DeletePartition()	Mandatory	See section 8.5
DCIM_PersistentStorageService.FormatPartition()	Mandatory	See section 8.6
DCIM_PersistentStorageService.ModifyPartition()	Mandatory	See section 8.7
DCIM_PersistentStorageService.AttachPartition()	Mandatory	See section 8.8
DCIM_PersistentStorageService.DetachPartition()	Mandatory	See section 8.9
DCIM_PersistentStorageService.ExportDataFromPartition()	Mandatory	See section 8.10

7.2.3   Properties

The following table details the implemented properties for DCIM_OpaqueManagementData instance representing a partition on media storage. The "Requirement" column shall denote the implementation requirement for the corresponding property. If the column "Properties" matches the property name, the property shall have the value denoted in the corresponding column "Additional Requirements".

Table 6 – Class: DCIM_OpaqueManagementData

Properties	Requirement	Type	Additional Requirements
SystemCreationClassName	Mandatory	string	The property value shall be "DCIM_ComputerSystem".
CreationClassName	Mandatory	string	The property value shall be "DCIM_OpaqueManagementData".
SystemName	Mandatory	string	The property value shall be "DCIM:ComputerSystem".
DeviceID	Mandatory	string	The property value shall be "DCIM_OpaqueManagementData:Partition<index> where <index> is equal to partition index.
AccessType	Mandatory	string	Access describes whether the media is readable. The property shall have value of: "Read Only" or "Read-Write".
AttachedState	Mandatory	string	This property represents if the partition is attached to OS.
DataFormat	Mandatory	string	The property shall have value of: "Raw", "Ext2", "Ext3", "FAT16", or "FAT32".
ElementName	Mandatory	string	The property value shall be "VFlash".
Name	Mandatory	string	A unique identifier for the partition.
PartitionIndex	Mandatory	uint16	This property represents the index of the partition.
PartitionType	Mandatory	string	This property represents the type of the partition which is specified at creation time. The property shall have value of: "Floppy", "HDD" or "CD-DVD".
Size	Mandatory	uint32	The property value shall be in MB.

7.3 DCIM_PersistentStorageService

This section describes the implementation for the DCIM_PersistentStorageService class.

This class shall be instantiated in the Implementation Namespace.

The DCIM_LCElementConformsToProfile association(s) shall reference the DCIM_PersistentStorageService 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_PersistentStorageService?__cimnamespace=root/dcim"

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

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

7.3.2 Operations

The following table details the implemented operations on DCIM_PersistentStorageService.

Table 7 – DCIM_PersistentStorageService – Operations

Operation Name	Requirements	Required Input

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

7.3.3   Properties

The following table details the implemented properties for DCIM_PersistentStorageService instance representing a system in a system. The "Requirement" column shall denote the implementation requirement for the corresponding property. If the column "Properties and Methods" matches the property name, the property shall have the value denoted in the corresponding column "Description".

Table 8 – Class: DCIM_PersistentStorageService

Properties and Methods	Requirement	Description
SystemCreationClassName	Mandatory	The property value shall be "DCIM_ComputerSystem".
CreationClassName	Mandatory	The property value shall be "DCIM_PersistentStorageService".
SystemName	Mandatory	The property value shall be "DCIM:ComputerSystem".
Name	Mandatory	The property value shall be "DCIM:PersistentStorageService".
ElementName	Mandatory	The property value shall be "Persistent Storage Service".

7.4   Persistent Storage Profile Registration

This section describes the implementation for the DCIM_LCRegisteredProfile class.

This class shall be instantiated in the Interop Namespace.

The DCIM_LCElementConformsToProfile association(s) 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:PersistentStorage:1.0.0"

7.4.2   Operations

The following table details the implemented operations on DCIM_LCRegisteredProfile.

Table 9 – DCIM_SystemView - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI

Operation Name	Requirements	Required Input
Enumerate	Mandatory	Class URI

7.4.3 Properties

The following table details the implemented properties for DCIM_LCRegisteredProfile instance representing Persistent Storage Profile implementation. The "Requirement" column shall denote the implementation requirement for the corresponding property. If the column "Properties" matches the property name, the property shall have the value denoted in the corresponding column "Description".

Table 10 – Class: DCIM_LCRegisteredProfile

Properties	Requirement	Type	Description
RegisteredName	Mandatory	String	This property shall have a value of "Persistent Storage".
RegisteredVersion	Mandatory	String	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	Uint16	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	Mandatory	String	This property shall match "DCIM"
InstanceID	Mandatory	String	This property shall have a value of "DCIM:PersistentStorage:1.0.0"
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_PersistentStorageService.InitializeMedia()

The InitializeMedia() method is used to initialize or format the virtual flash media device.

Upon successful invocation of the InitializeMedia() method, the DCIM_VFlashView.InitializedState property shall have string value "Initialized".

Return code values for the InitializeMedia( ) method are specified in Table 11 and parameters are specified in Table 12.

Table 11 – InitializeMedia( ) Method: Return Code Values

Value	Description
2	Failed
4096	Job Created

Table 12 – InitializeMedia( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT	Job	CIM_Concrete Job REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 13 – InitializeMedia( ) Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
VF001	The command was successful	
VF002	General failure	
VF015	VFlash not enabled	

8.2   DCIM_PersistentStorageService.VFlashStateChange( )

The VFlashStateChange( ) method is used to enable or disable the virtual flash media device.

Upon successful invocation of the VFlashStateChange( ) method with RequestedState 1(Enable), the DCIM_VFlashView. VFlashEnabledState property value shall change to TRUE.

Upon successful invocation of the VFlashStateChange( ) method with RequestedState 2(Disable), the DCIM_VFlashView. VFlashEnabledState property value shall change to FALSE.

Return code values for the VFlashStateChange( ) method are specified in Table 14, and parameters are specified in Table 15.

Table 14 – VFlashStateChange( ) Method: Return Code Values

Value	Description
0	Completed with no error
2	Failed

Table 15 – VFlashStateChange( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	Uint32	Shall be set to: 1 (Enable) or 2 (Disable)
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 16 – VFlashStateChange( ) Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
VF001	The command was successful	
VF002	General failure	
VF003	Missing required parameter <parameter>	RequestedState
VF004	Invalid value of parameter <parameter>	RequestedState
VF014	VFlash not disabled	
VF013	SD card locked	
VF024	Partition locked	
VF027	Partition already attached	
VF052	SD card not ready	
VF014	VFlash not disabled	
VF01	SD card locked	
VF024	Partition locked	

8.3   DCIM_PersistentStorageService.CreatePartition( )

The CreatePartition( ) method is used to used for creating a new partition on a storage device.

Upon successful invocation of the CreatePartition() method, a new instance of DCIM_OpaqueManagementData shall be created with the PartitionIndex property equal to the PartitionIndex parameter, the Size property in MB equal to the Size parameter in the SizeUnit parameter specified units, the PartitionType property equal to the string representation of PartitionType parameter value map, and the Name property equal to the OSVolumeLabel parameter.

Return code values for the CreatePartition( ) method are specified in Table 17, and parameters are specified in Table 18.

Table 17 – CreatePartition( ) Method: Return Code Values

Value	Description
2	Failed
4096	Job Created

Table 18 – CreatePartition( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.
IN, REQ	Size	Uint16	The size of the partition that needs to be created in units specified by the SizeUnit parameter.
IN, REQ	SizeUnit	Uint16	The value shall be: 1(MB) or 2(GB).
IN, REQ	PartitionType	Uint16	The value shall be: 1(Floppy) or 2(Hard Disk).
IN, REQ	OSVolumeLabel	String	The value shall be limited to 6 characters. This is the same label which will be seen in the OS after attaching the partition.
OUT	Job	CIM_ConcreteJob REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 19 – CreatePartition( ) Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP017	Maximum job limit reached	
JCP018	Cannot create new jobs until the existing Export job is completed or deleted.	
JCP019	Cannot create new jobs until the existing Import job is completed or deleted.	
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter <parameter>	OSVolumeLabel, PartitionIndex, PartitionType, SizeUnit, Size
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF016	SD card is read only	
VF017	SD card not initialized	
VF018	Partition does not exist	
VF019	Not enough space on SD card	
VF021	Exceeded maximum partition size	
VF022	Partition size below minimum	
VF023	Partition label not unique	
VF026	Partition index in use	
VF031	Invalid partition label	
VF032	Invalid partition type	
VF052	SD card not ready	

8.4 DCIM_PersistentStorageService.CreatePartitionUsingImage( )

The CreatePartitionUsingImage( ) method is used to create a partition using an image provided by the user. Image may reside on (T)FTP or HTTP server, or on a CIFS or NFS share. Upon successful invocation of the CreatePartitionUsingImage() method, the image shall be downloaded and provisioned as a partition on a VFlash.

Upon successful invocation of the CreatePartitionUsingImage() method, a new instance of DCIM_OpaqueManagementData shall be created with the PartitionIndex property equal to the PartitionIndex parameter, the Size property in MB equal to the size of the remote image, the PartitionType property equal to the string representation of PartitionType parameter value map, and the Name property equal to the OSVolumeLabel parameter.

Return code values for the CreatePartitionUsingImage( ) method are specified in Table 20, and parameters are specified in Table 21.

Table 20 – CreatePartitionUsingImage( ) Method: Return Code Values

Value	Description
2	Failed
4096	Job Created

Table 21 – CreatePartitionUsingImage( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.
IN, REQ	PartitionType	Uint16	The value shall be: 1(Floppy), 2(Hard Disk), or 3(CDROM)
IN, REQ	OSVolumeLabel	String	The value shall be limited to 6 characters. This is the same label which will be seen in the OS after attaching the partition.
IN	URI	String	The parameter shall be populated, if the ShareType parameter is set to 3(FTP), or 4 (HTTP)
IN	IPAddress	String	The parameter shall be populated, if the ShareType parameter is set to 0 (NFS) or, 1(TFTP), or 2(CIFS)
IN, REQ	ShareType	Uint16	0 (NFS), 1(TFTP), 2(CIFS), 3(FTP), or 4 (HTTP)
IN	SharePath	String	The parameter shall be populated, if the ShareType parameter is set to 0 (NFS) or, 1(TFTP), or 2(CIFS)
IN, REQ	ImageName	String	Name of the image to create the partition
IN	Workgroup	String	Applicable Workgroup
IN, REQ	Username	String	The username for accessing the image on a remote node.
IN, REQ	Password	String	The password for accessing the image on a remote node.
IN	Port	Uint16	The parameter shall be populated, if the ShareType parameter is set to 1(TFTP).
IN	HashType	Uint16	The value shall be: 1(MD5) or 2(SHA1)
IN	HashValue	String	The HashValue parameter shall be set to the hash value of the image using the specified hash type in the HashType parameter.
OUT	Job	CIM_ConcreteJob REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 22 – CreatePartitionUsingImage( ) Method:Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP017	Maximum job limit reached	
JCP018	Cannot create new jobs until the existing Export job is completed or deleted.	
JCP019	Cannot create new jobs until the existing Import job is completed or deleted.	
VF001	The command was successful	
VF002	General failure	

MessageID (OUT parameter)	Message	MessageArguments[]
VF004	Invalid value of parameter <parameter>	OSVolumeLabel, PartitionIndex, PartitionType, HashType, ShareType
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF016	SD card is read only	
VF019	Not enough space on SD card	
VF021	Exceeded maximum partition size	
VF022	Partition size below minimum	
VF023	Partition label not unique	
VF031	Invalid partition label	
VF032	Invalid partition type	
VF040	Image hash verification failed	
VF041	Remote mount failed	
VF042	Remote file missing	
VF043	Remote file too large	
VF044	TFTP download failed	
VF045	FTP download failed	
VF046	HTTP download failed	
VF052	SD card not ready	

8.5 DCIM_PersistentStorageService.DeletePartition( )

The DeletePartition( ) method is used for deleting a partition on a vFlash.

Upon the successful execution of the DeletePartition( ) method, the DCIM_OpaqueManagementData instance with PartitionIndex property equal to the PartitionIndex parameter shall be deleted.

Return code values for the DeletePartition( ) method are specified in Table 23 and parameters are specified in Table 24.

Table 23 – DeletePartition( ) Method: Return Code Values

Value	Description
0	Completed with no error
2	Failed

Table 24 – DeletePartition( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message

Qualifiers	Name	Type	Description/Values
OUT	MessageArguments[]	String	Error MessageArguments

<p align="center">Table 25 – DeletePartition( ) Method:Standard Messages</p>

MessageID (OUT parameter)	Message	MessageArguments[]
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter %s	PartitionIndex
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF017	SD card not initialized	
VF024	Partition locked	
VF026	Partition index in use	
VF027	Partition already attached	
VF047	Unsupported SD card	
VF052	SD card not ready	

8.6   DCIM_PersistentStorageService.FormatPartition( )

The FormatPartition( ) method is used for formatting a partition on a vFlash.

Upon the successful execution of the FormatPartition( ) method, the DCIM_OpaqueManagementData instance with PartitionIndex property equal to the PartitionIndex parameter shall be formatted to the format type specified in the FormatType parameter, and the DataFormat property shall have the string value of the FormatType parameter's value map.

Return code values for the FormatPartition( ) method are specified in Table 26 and parameters are specified in Table 27.

<p align="center">Table 26 – FormatPartition( ) Method: Return Code Values</p>

Value	Description
2	Failed
4096	Job Created

<p align="center">Table 27 – FormatPartition( ) Method: Parameters</p>

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.
IN, REQ	FormatType	Uint16	The value shall be: 1(EXT2), 2(EXT3), 3(FAT16) or 4(FAT 32)
OUT	Job	CIM_Concrete Job REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 28 – FormatPartition( ) Method:Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP017	Maximum job limit reached	
JCP018	Cannot create new jobs until the existing Export job is completed or deleted.	
JCP019	Cannot create new jobs until the existing Import job is completed or deleted.	
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter <parameter>	PartitionIndex, FormatType
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF016	SD card is read only	
VF018	Partition does not exist	
VF019	Not enough space on SD card	
VF021	Exceeded maximum partition size	
VF024	Partition locked	
VF026	Partition index in use	
VF027	Partition already attached	
VF033	Invalid partition format type	
VF047	Unsupported SD card	
VF052	SD card not ready	

8.7 DCIM_PersistentStorageService.ModifyPartition( )

The ModifyPartition( ) method is used for modifying a partition on a vFlash.

Upon the successful execution of the ModifyPartition( ) method, the partition, represented by the DCIM_OpaqueManagementData instance with the PartitionIndex property equal to the PartitionIndex parameter, shall be modified to the access type specified in the AccessType parameter.

Return code values for the ModifyPartition( ) method are specified in Table 29 and parameters are specified in Table 30.

Table 29 – ModifyPartition( ) Method: Return Code Values

Value	Description
0	Completed with no error
2	Failed

Table 30 – ModifyPartition( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.

Qualifiers	Name	Type	Description/Values
IN, REQ	AccessType	Uint16	The value shall be: 1 (Read-Only), or 3(Read-Write)
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 31 – ModifyPartition( ) Method:Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter <parameter>	PartitionIndex, AccessType
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF016	SD card is read only	
VF017	SD card not initialized	
VF018	Partition does not exist	
VF024	Partition locked	
VF026	Partition index in use	
VF028	Partition already detached	
VF034	Invalid partition access type	
VF047	Unsupported SD card	
VF052	SD card not ready	

8.8 DCIM_PersistentStorageService.AttachPartition( )

The AttachPartition( ) method is used to expose a partition as to the managed system's OS.

Upon the successful execution of the AttachPartition( ) method, the partition, represented by the DCIM_OpaqueManagementData instance with the PartitionIndex property equal to the PartitionIndex parameter, shall be exposed to the host OS.

Return code values for the AttachPartition( ) method are specified in Table 32 and parameters are specified in Table 33.

Table 32 – AttachPartition( ) Method: Return Code Values

Value	Description
2	Failed
4096	Job Created

Table 33 – AttachPartition( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.

Qualifiers	Name	Type	Description/Values
OUT	Job	CIM_Concrete Job REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 34 – AttachPartition( ) Method:Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP017	Maximum job limit reached	
JCP018	Cannot create new jobs until the existing Export job is completed or deleted.	
JCP019	Cannot create new jobs until the existing Import job is completed or deleted.	
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter %s	PartitionIndex
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF016	SD card is read only	
VF017	SD card not initialized	
VF018	Partition does not exist	
VF024	Partition locked	
VF026	Partition index in use	
VF047	Unsupported SD card	
VF052	SD card not ready	

8.9 DCIM_PersistentStorageService.DetachPartition( )

The DetachPartition( ) method is used to remove a partition that was previously exposed to the managed system's OS.

Upon the successful execution of the DetachPartition( ) method, the partition, represented by the DCIM_OpaqueManagementData instance with the PartitionIndex property equal to the PartitionIndex parameter, shall be removed from the host OS.

Return code values for the DetachPartition( ) method are specified in Table 35 and parameters are specified in Table 36.

Table 35 – DetachPartition( ) Method: Return Code Values

Value	Description
2	Failed
4096	Job Created

Table 36 – DetachPartition( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.
OUT	Job	CIM_Concrete Job REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 37 – DetachPartition( ) Method:Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP017	Maximum job limit reached	
JCP018	Cannot create new jobs until the existing Export job is completed or deleted.	
JCP019	Cannot create new jobs until the existing Import job is completed or deleted.	
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter %s	PartitionIndex
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	
VF016	SD card is read only	
VF017	SD card not initialized	
VF018	Partition does not exist	
VF024	Partition locked	
VF026	Partition index in use	
VF047	Unsupported SD card	
VF052	SD card not ready	

8.10 DCIM_PersistentStorageService.ExportDataFromPartition( )

The ExportDataFromPartition( ) method is used to export an image from a partition and transfer it to a (T)FTP or HTTP server, or on a CIFS or NFS share. Upon successful invocation of the ExportDataFromPartition() method, the image of the partition shall be exported from a VFlash to a remote endpoint.

Return code values for the ExportDataFromPartition( ) method are specified in Table 38, and parameters are specified in Table 39.

Table 38 – ExportDataFromPartition( ) Method: Return Code Values

Value	Description
2	Failed

Value	Description
4096	Job Created

Table 39 – ExportDataFromPartition( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	PartitionIndex	Uint16	The index of the partition that shall have value be between 1 and 16.
IN, REQ	IPAddress	String	The parameter shall be populated, if the ShareType parameter is set to 0 (NFS) or, 1(TFTP), or 2(CIFS)
IN, REQ	ShareType	Uint16	0 (NFS), 1(TFTP), 2(CIFS), 3(FTP), or 4 (HTTP)
IN, REQ	SharePath	String	The parameter shall be populated, if the ShareType parameter is set to 0 (NFS) or, 1(TFTP), or 2(CIFS)
IN, REQ	ImageName	String	Name of the ISO or IMG image
IN	Workgroup	String	Applicable Workgroup
IN, REQ	Username	String	The username for accessing the image on a remote node.
IN, REQ	Password	String	The password for accessing the image on a remote node.
IN	Port	Uint16	The parameter shall be populated, if the ShareType parameter is set to 1(TFTP).
OUT	Job	CIM_ConcreteJob REF	Returned if job is created.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

Table 40 – ExportDataFromPartition( ):Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP017	Maximum job limit reached	
JCP018	Cannot create new jobs until the existing Export job is completed or deleted.	
JCP019	Cannot create new jobs until the existing Import job is completed or deleted.	
VF001	The command was successful	
VF002	General failure	
VF004	Invalid value of parameter %s	PartitionIndex,IPAddress, ShareType, SharePath, ImageName, Workgroup, Username, Password, Port
VF012	SD card not present	
VF013	SD card locked	
VF015	VFlash not enabled	

MessageID (OUT parameter)	Message	MessageArguments[]
VF016	SD card is read only	
VF017	SD card not initialized	
VF018	Partition does not exist	
VF018	Partition does not exist	
VF024	Partition locked	
VF041	Remote mount failed	
VF042	Remote file missing	
VF043	Remote file too large	
VF044	TFTP download failed	
VF045	FTP download failed	
VF047	Unsupported SD card	
VF052	SD card not ready	

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 describes 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 41 – Privilege and License Requirements

Class and Method	Operation	User Privilege Required	License Required
DCIM_OpaqueManagementData	ENUMERATE, GET	Login	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_VFlashView	ENUMERATE, GET	Login	LM_REMOTE_ASSET_INVE NTORY
DCIM_PersistentStorageService	ENUMERATE, GET	Login	None.
DCIM_PersistentStorageService. InitializeMedia()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. VFlashStateChange()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. CreatePartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS

Class and Method	Operation	User Privilege Required	License Required
DCIM_PersistentStorageService. CreatePartitionUsingImage()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. DeletePartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. FormatPartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. ModifyPartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. AttachPartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. DetachPartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_PersistentStorageService. ExportDataFromPartition()	INVOKE	Login, Virtual Media	LM_VIRTUAL_FLASH_PART ITIONS
DCIM_LCRegisteredProfile	ENUMERATE, GET	Login	None.
DCIM_LCElementConformsToProfile	ENUMERATE, GET	Login	None.