DCIM Operating System (OS) Deployment Profile

Document Number: DCIM1035
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2011-03-22

Version: 1.2.0

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

Operating System (OS) Deployment Profile

1 Scope

The Operating System (OS) Deployment Profile extends the management capabilities of referencing profiles by adding the capability to represent OS deployment configuration features. The OS deployment feature consists of:

- Provide the supported OS and version information from the embedded OS driver pack.
- Unpack the OS Driver Update Package (DUP) and extract drivers for a specified OS; the drivers are placed on a local flash drive (nvram) or on a network share.
- Expose the unpacked drivers to the host as a USB device.
- Ability to boot to PXE images.
- Ability to boot to the ISO image present on a network share.
- Ability to download ISO Image and save it in vFlash.
- Ability to boot to the ISO present in vFlash.

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 DSP1033, *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*

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

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

4 Symbols and Abbreviated Terms

4.1

CIM

Common Information Model

5 Synopsis

Profile Name: Operating System (OS) Deployment

Version: 1.2.0

Organization: DCIM

CIM Schema Version: 2.19.1

Central Class: DCIM_OSDeploymentService

Scoping Class: CIM_ComputerSystem

The **Operating System (OS) Deployment Profile** extends the management capability of the referencing profiles by adding the capability to support OS deployment activities by manipulating the OS deployment features provided by the service processor. These features include:

- Identify the list of OS drivers.
- Unpack the OS DUP and extract drivers for a specified OS; the drivers are placed on a local flash drive (nvram).
- Expose the unpacked drivers to the host as a USB device.
- Expose the drivers to the network in a secure manner for access by a management application.
- Support booting to an OS provisioning image from a network share.
- Download a pre-boot OS provisioning image from a network share to vFlash.
- Support booting to an OS provisioning image from vFlash.

DCIM_OSDeploymentService shall be the Central Class.

Table 1 identifies related profiles.

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration Profile	DMTF	1.0	Mandatory

6 Description

The Operating System (OS) Deployment Profile describes the OS deployment configuration service, and the methods exposed to manage the service processor OS deployment features. The profile also describes the relationship of the OS Deployment Profile classes to DMTF and Dell profile version information.

The interface for the OS deployment feature consists of the following functionality:

- Identify the list of OS drivers.
- Unpack the OS DUP and extract drivers for a specified OS; the drivers are placed on a local flash drive (nvram).
- Expose the unpacked drivers to the host as a USB device.
- Expose the drivers to the network in a secure manner for access by a management application.
- Support booting to an OS provisioning image from a network share.
- Download a pre-boot OS provisioning image from a network share to vFlash.
- Support booting to an OS provisioning image from vFlash.

Figure 1 represents the class schema for the **Operating System (OS) Deployment Profile**. For simplicity, the prefix CIM_ has been removed from the names of the classes that are standard DMTF classes.

The OS Deployment feature in a service processor is represented by the instance of the DCIM_OSDeploymentService class. The DCIM_OSDeploymentService has extrinsic methods for accomplishing the various OS deployment features described above. The OS Deployment profile information is represented with the instance of CIM_RegisteredProfile.

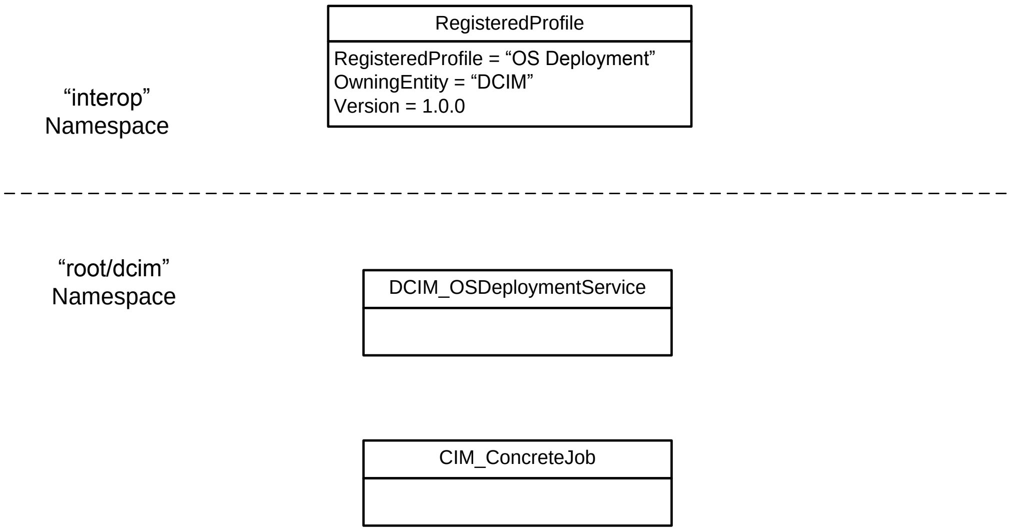

Figure 1 – Operating System (OS) Deployment Profile**: Class Diagram**

7 Implementation Requirements

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

7.1 DCIM_OSDeploymentService

One instance of DCIM_OSDeploymentService shall be instantiated.

7.1.1 ElementName

The value of ElementName shall be formulated using the following pattern:

ElementName = “OSD”

7.2 CIM_ConcreteJob

At most one instance of CIM_ConcreteJob shall be instantiated. An instance of CIM_ConcreteJob may be returned as an output parameter for all the extrinsic methods supported by DCIM_OSDeploymentService

7.2.1 Name

The value of Name shall be formulated using the following pattern:

Name = “DCIM_OSDeploymentService.ExtrinsicMethodName”, ex: Name=”UnpackAndAttach”

CIM_ConcreteJob reference returned as an output of DCIM_OSDeploymentService.UnpackAndAttach, shall have the following Name:

- CIM_ConcreteJob.Name = “UnpackAndAttach”

CIM_ConcreteJob reference returned as an output of DCIM_OSDeploymentService.UnpackAndShare, shall have the following Name:

- CIM_ConcreteJob.Name = “UnpackAndShare”.

CIM_ConcreteJob reference returned as an output of DCIM_OSDeploymentService.BootToNetworkISO, shall have the following Name:

- CIM_ConcreteJob.Name = “BootToNetworkISO”.

7.2.2 JobStatus

The value of the JobStatus, a free form string, property shall be one of the entries in Table 2

Table 2 – JobStatus

Job Name	JobStatus	JobStatus Description
“UnpackAndAttach”	Processing Driver Pack	Extracting drivers, creating dynamic partition, copying drivers, and attaching the partition as a USB device to the host.
“UnpackAndAttach”	Success	Successfully executed the method.
“UnpackAndAttach”	Failed	Failed to execute the method, refer to MessageID and Message properties of the DCIM_OSDConcreteJob instance for detailed information.
“UnpackAndShare”	Processing Driver Pack	Extracting drivers and copying drivers to

Job Name	JobStatus	JobStatus Description
		the network share.
“UnpackAndShare”	Success	Successfully executed the method.
“UnpackAndShare”	Failed	Failed to execute the method, refer to MessageID and Message properties of the DCIM_OSDConcreteJob instance for detailed information.
BootToNetworkISO	Rebooting to ISO	Attaching the network ISO image as a local CDROM to the host and booting to it.
BootToNetworkISO	Success	Successfully executed the method.
BootToNetworkISO	Failed	Failed to execute the method, refer to MessageID and Message properties of the DCIM_OSDConcreteJob instance for detailed information.
BootToISOFromVFlash	Rebooting to ISO	Attaching the ISO image on vFlash as a local CDROM to the host and booting to it.
BootToISOFromVFlash	Success	Successfully executed the method.
BootToISOFromVFlash	Failed	Failed to execute the method, refer to MessageID and Message properties of the DCIM_OSDConcreteJob instance for detailed information.
DownloadISOToVFlash	Downloading	Copying the ISO image from network share to vFlash.
DownloadISOToVFlash	Success	Successfully executed the method.
DownloadISOToVFlash	Failed	Failed to execute the method, refer to MessageID and Message properties of the DCIM_OSDConcreteJob instance for detailed information.
ConnectNetworkISOImage	Connecting to Network ISO	Connecting to the ISO present on network share and attaching it as a CDROM device to the host
ConnectNetworkISOImage	Success	Successfully executed the method.
ConnectNetworkISOImage	Failed	Failed to execute the method, refer to MessageID and Message properties of the DCIM_OSDConcreteJob instance for detailed information.

8 Methods

This section details the requirements for supporting extrinsic methods for the CIM elements defined by this profile.

8.1 Method: DCIM_OSDeploymentService.GetDriverPackInfo()

The GetDriverPackInfo method returns the list of OSs that can be installed on the server using the embedded device drivers present in the Lifecycle Controller.

Return values for GetDriverPackInfo() shall be as specified in Table 3 where the method-execution behavior matches the return-code description. GetDriverPackInfo() method’s parameters are specified in Table 4.

Output parameters are Version and OSList.

Table 3 – DCIM_OSDeploymentService.GetDriverPackInfo() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 4 – DCIM_OSDeploymentService.GetDriverPackInfo() Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (required)	Version	String	NULL or version of the driver pack.
OUT (required)	OSList	String[]	NULL or contains the list of operating systems supported for this server.
OUT (optional)	Job	CIM_ConcreteJob	NULL or reference to a CIM_ConreteJob
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.2 Method: DCIM_OSDeploymentService.UnpackAndAttach()

The UnpackAndAttach() method will extract the drivers for the selected OS to a USB device that will be attached locally to the server for the specified time interval. Return values for UnpackAndAttach() shall be as specified in Table 5, where the method-execution behavior matches the return-code description. UnpackAndAttach( ) method’s parameters are specified in Table 6.

No standard messages are defined for this method.

Table 5 – DCIM_OSDeploymentService.UnpackAndAttach() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 6 – DCIM_OSDeploymentService.UnpackAndAttach() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	OSName	String	Name of the OS to unpack drivers for, this value shall match one of the strings in OSList returned for GetDriverPackInfo
IN	ExposeDuration	DateTime	Identifies the amount of time up to 18 hours for the drivers to be exposed as an USB device to the host.
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – USB attach job.
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.3 Method: DCIM_OSDeploymentService.DetachDrivers()

This method will detach the USB device containing the drivers from the host server. Return values for DetachDrivers() shall be as specified in Table 7, where the method-execution behavior matches the return-code description. DetachDrivers() method’s parameters are specified in Table 8.

Table 7 – DCIM_OSDeploymentService.DetachDrivers() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred.

Table 8 – DCIM_OSDeploymentService.DetachDrivers() Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.4 Method: DCIM_OSDeploymentService.UnpackAndShare()

The UnpackAndShare method will extract the drivers for the selected OS, and copy them to the specified network share. Return values for UnpackAndShare() shall be as specified in Table 9, where the method-execution behavior matches the return-code description. UnpackAndShare ( ) method’s parameters are specified in Table 10.

No standard messages are defined for this method.

Table 9 – DCIM_OSDeploymentService.UnpackAndShare() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 10 – DCIM_OSDeploymentService.UnpackAndShare() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	IPAddress	String	CIFS or NFS share IPv4 address. Example 192.168.10.100
IN	ShareName	String	NFS/CIFS share name. Here is an example for share name. NFS share name - "/home/guest". CIFS share name - “guest_smb”
IN	OSName	String	OS name
IN	ShareType	Uint32	0=NFS | CIFS = 2
IN	Workgroup	String	Workgroup name, if applicable
IN (optional)	UserName	String	User name, if applicable
IN (optional)	Password	String	Password, if applicable
IN (optional)	Port	Uint32	Port number, if applicable
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – UnpackAndShare job.
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.5 Method: DCIM_OSDeploymentService.BootToNetworkISO()

The BootToNetworkISO() method is used to boot to the iDRAC pre-OS image that was already downloaded. Return values for BootToNetworkISO () shall be as specified in Table 11, where the method-execution behavior matches the return-code description. BootToNetworkISO( ) method’s parameters are specified in Table 12. No standard messages are defined for this method.

Table 11 – DCIM_OSDeploymentService.BootToNetworkISO() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 12 – DCIM_OSDeploymentService.BootToNetworkISO() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	IPAddress	String	NFS or CIFS share IPv4 address. Example 192.168.10.100
IN	ShareName	String	NFS or CIFS share name. Here is an example for share name. NFS share name - "/home/guest". CIFS share name - “guest_smb”
IN	ImageName	String	ISO image name
IN	ShareType	Uint32	0=NFS | CIFS = 2
IN	Workgroup	String	Workgroup name, if applicable
IN (optional)	UserName	String	User name, if applicable
IN (optional)	Password	String	Password, if applicable
IN (optional)	Port	Uint32	Port number, if applicable
IN (optional)	ExposeDuration	DateTime	Identifies the amount of time (up to 18 hours) for the ISO Image to be exposed as a local CD-ROM device to the host.
IN (optional)	HashType	Uint16	Type of hash algorithm used to compute checksum (1=MD5 | 2=SHA1)
IN (optional)	HashValue	String	Checksum value in string format computed using the HashType algorithm
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – UnpackAndShare job.
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.6 Method: DCIM_OSDeploymentService.DetachISOImage()

This method will detach the ISO Image from the host server. Return values for DetachISOImage() shall be as specified in Table 13, where the method-execution behavior matches the return-code description. DetachISOImage () method’s parameters are specified in Table 14.

Table 13 – DCIM_OSDeploymentService.DetachISOImage() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred.

Table 14 – DCIM_OSDeploymentService.DetachISOImage() Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.7 Method: DCIM_OSDeploymentService.BootToPXE()

The BootToPXE() method is used to Boot to server using the PXE mechanism. Return values for BootToPXE () shall be as specified in Table 15, where the method-execution behavior matches the return-code description. BootToPXE( ) method's parameters are specified in Table 16. No standard messages are defined for this method.

Table 15 – DCIM_OSDeploymentService.BootToPXE() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 16 – DCIM_OSDeploymentService.BootToPXE() Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.8 Method: DCIM_OSDeploymentService.DownloadISOToVFlash()

The DownloadISOToVFlash() method is used to download the pre-OS ISO Image to vFlash. Return values for DownloadISOToVFlash() shall be as specified in Table 17, where the method-execution behavior matches the return-code description. DownloadISOToVFlash( ) method's parameters are specified in Table 18. No standard messages are defined for this method.

Table 17 – DCIM_OSDeploymentService.DownloadISOToVFlash() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 18 – DCIM_OSDeploymentService.DownloadISOToVFlash() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	IPAddress	String	TFTP, CIFS, or NFS share IPv4 address. Example 192.168.10.100
IN	ShareName	String	Network share point. Here is an example for share name. NFS share name - "/home/guest". CIFS share name - “guest_smb”
IN	ImageName	String	ISO Image name
IN	ShareType	Uint32	0=NFS | 1=TFTP | CIFS = 2
IN(optional)	Workgroup	String	Workgroup name, if applicable
IN (optional)	UserName	String	User name, if applicable
IN (optional)	Password	String	Password, if applicable
IN (optional)	Port	Uint32	Port number, if applicable
IN (optional)	HashType	Uint16	Type of hash algorithm used to compute checksum (1=MD5 | 2=SHA1)
IN (optional)	HashValue	String	Checksum value in string format computed using HashType algorithm
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – UnpackAndShare job.
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.9 Method: DCIM_OSDeploymentService.BootToISOFromVFlash()

The BootToISOFromVFlash() method is used to boot to the vFlash pre-OS image that was already downloaded. Return values for BootToISOFromVFlash() shall be as specified in Table 19, where the method-execution behavior matches the return-code description. BootToISOFromVFlash( ) method’s parameters are specified in Table 20. No standard messages are defined for this method.

Table 19 – DCIM_OSDeploymentService.BootToISOFromVFlash() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 20 – DCIM_OSDeploymentService.BootToISOFromVFlash() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN (optional)	ExposeDuration	DateTime	Identifies the amount of time (up to 18 hours) for the ISO image to be exposed as a local CD-ROM device to the host.
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – UnpackAndShare job.
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.10 Method: DCIM_OSDeploymentService.DetachISOFromVFlash()

This method will detach the ISO Image from the host server.

Return values for DetachISOFromVFlash() shall be as specified in Table 21 where the method-execution behavior matches the return-code description. DetachISOFromVFlash() method’s parameters are specified in Table 22.

Table 21 – DCIM_OSDeploymentService.DetachISOFromVFlash() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred.

Table 22 – DCIM_OSDeploymentService.DetachISOFromVFlash() Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.11 Method: DCIM_OSDeploymentService.DeleteISOFromVFlash()

This method will delete the ISO Image from vFlash. Return values for DeleteISOFromVFlash() shall be as specified in Table 23, where the method-execution behavior matches the return-code description. DeleteISOFromVFlash() method’s parameters are specified in Table 24.

Table 23 – DCIM_OSDeploymentService.DeleteISOFromVFlash() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred.

Table 24 – DCIM_OSDeploymentService.DeleteISOFromVFlash() Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.12 Method: DCIM_OSDeploymentService.ConnectNetworkISOImage()

This method will connect to the ISO present on the network share and expose the ISO as a local USB CDROM device to the host server
Return values for ConnectNetworkISOImage () shall be as specified in Table 25 where the method-execution behavior matches the return-code description. ConnectNetworkISOImage() method’s parameters are specified in Table 26.

Table 25 – DCIM_OSDeploymentService.ConnectNetworkISOImage () Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.
4096	Job started: REF returned to started CIM_ConcreteJob

Table 26 – DCIM_OSDeploymentService.ConnectNetworkISOImage () Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	IPAddress	String	CIFS, or NFS share IPv4 address. Example 192.168.10.100
IN	ShareName	String	Network share point. Here is an example for share name. NFS share name - "/home/guest". CIFS share name - “guest_smb”
IN	ImageName	String	ISO Image name
IN	ShareType	Uint32	0=NFS | CIFS = 2
IN(optional)	Workgroup	String	Workgroup name, if applicable
IN (optional)	UserName	String	User name, if applicable
IN (optional)	Password	String	Password, if applicable
IN (optional)	HashType	Uint16	Type of hash algorithm used to compute checksum (1=MD5 | 2=SHA1)
IN (optional)	HashValue	String	Checksum value in string format computed using HashType algorithm
IN (optional)	AutoConnect	Boolean	Auto-connect to ISO Image upon iDRAC reset
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – UnpackAndShare job.
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.13 Method: DCIM_OSDeploymentService.DisconnectNetworkISOImage ()

This method will disconnect and detach the ISO Image from the host server
Return values for DisconnectNetworkISOImage () shall be as specified in Table 27 where the method-execution behavior matches the return-code description. DisconnectNetworkISOImage () method’s parameters are specified in Table 28.

Table 27 – DCIM_OSDeploymentService.DisconnectNetworkISOImage () Method: Return Code Values

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

Table 28 – DCIM_OSDeploymentService.DisconnectNetworkISOImage () Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.14 Method: DCIM_OSDeploymentService.GetNetworkISOImageConnectionInfo()

This method will give the status of the ISO Image that has been exposed to host
Return values for GetNetworkISOImageConnectionInfo () shall be as specified in Table 29 where the method-execution behavior matches the return-code description. GetNetworkISOImageConnectionInfo() method's parameters are specified in Table 30.

Table 29 – DCIM_OSDeploymentService.GetNetworkISOImageConnectionInfo() Method: Return Code Values

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

Table 30 – DCIM_OSDeploymentService. GetNetworkISOImageConnectionInfo () Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT	IPAddress	String	CIFS, or NFS share IPv4 address. Example 192.168.10.100
OUT	ShareName	String	Network share point. Here is an example for share name. NFS share name - "/home/guest". CIFS share name - “guest_smb”
OUT	ImageName	String	ISO Image name
OUT	ShareType	Uint32	0=NFS | CIFS = 2
OUT (optional)	Workgroup	String	Workgroup name, if applicable
OUT	ISOConnectionStatus	Uint8	Describes if the ISO connection status. If the ISO is still accessible or not.
OUT	HostAttachedStatus	Uint8	Describes ISO attached status. If the ISO is attached to the host server or not.
OUT	HostBootedFromISO	Uint8	Describes ISO boot status. If the ISO has been booted atleast once or not
OUT (optional)	UserName	String	User name, if applicable
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

8.15 Method: DCIM_OSDeploymentService.SkipISOImageBoot ()

This method will allow BIOS to skip booting to the ISO once and boot normally (boot to the first device in boot list)
Return values for SkipISOImageBoot () shall be as specified in Table 31 where the method-execution behavior matches the return-code description. SkipISOImageBoot () method’s parameters are specified in Table 32.

Table 31 – DCIM_OSDeploymentService.SkipISOImageBoot () 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_OSDeploymentService.SkipISOImageBoot () Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT (optional)	MessageID	String	If the method fails to execute, the error message ID is returned.
OUT (optional)	Message	String	If the method fails to execute, the error message in English is returned.
OUT (optional)	MessageArguments	string[]	Substitution variables for dynamic error messages

9 Use Cases

This section contains object diagrams and use cases for the OS Deployment Profile.

9.1 Object Diagram

Figure 2 represents a possible instantiation of the **Operating System (OS) Deployment Profile**, including advertising the profile. The object diagram in Figure 2 shows how an instance of CIM_RegisteredProfile is used to identify the version of OS Deployment Profile with an instance of DCIM_OSDeploymentService, and its associated instances are conformant. For simplicity, the prefix CIM_ has been removed from the names of the standard classes in the figure.

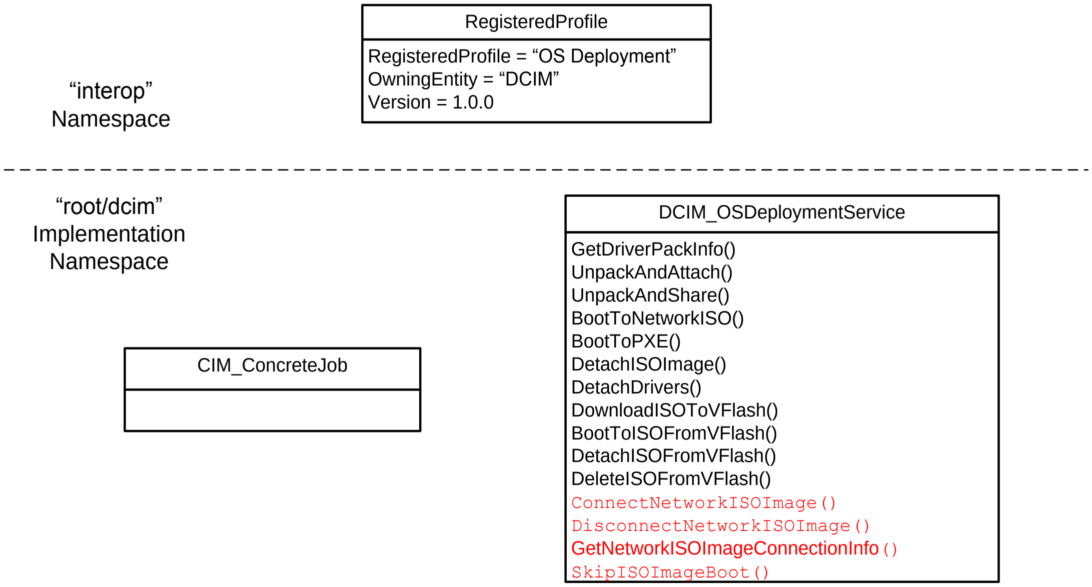

Figure 2 – Operating System (OS) Deployment Profile**: Object Diagram**

9.2 Discover OSD profile

Following sequence of CIM operations shall be used to discover the implemented version of OSD profile.

1. Enumerate (namespace=’root/interop’, classname=”CIM_RegisteredProfile”)
2. Filter the returned enumeration using the property filter (RegisteredProfile=”OS Deployment”)
3. Result shall contain one instance of CIM_RegisteredProfile containing property version=”1.0.0”

9.3 Get driver pack version and supported OS information

Following sequence of CIM Operations shall be used to retrieve the driver pack version, and supported OSs for OS deployment.

1. Follow the steps (1,2,3) from 9.2
2. Associators (objectpath= “instance returned from step 9.2.3”, resultclass=”DCIM_OSDeploymentSerive”) OR Enumerate (namespace=”root/dcim”, classname=”DCIM_OSDeploymentService”)
3. Result shall contain one instance of DCIM_OSDeploymentService
4. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “GetDriverPackInfo”
5. Invoke method shall return the following output parameters:
    a. Version = String version

    b. SupportedOperatingSystems = String array of OS names

    a. CIM_ConcreteJob

9.4 Unpack drivers and Attach to Host OS

Following sequence of CIM Operations shall be used to unpack drivers for the selected OS to a local partition, and attach the partition to the host OS.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “UnpackAndAttach”
    c. Refer to dcim_osdeploy.mof for rest of the method input parameters and data types
4. Invoke method shall return the following output parameters:
    a. Job = object path to CIM_ConcreteJob (reports the status of unpack and attach)

9.5 Unpack and share

Following sequence of CIM Operations shall be used to extract the drivers for the selected OS, and copy them to the specified network share.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “UnpackAndShare”
    c. Refer to dcim_osdeploy.mof for rest of the method input parameters and data types
4. Refer to the dcim_osdeploy.mof for rest of the method output parameters.

9.6 Boot to Network ISO image

Following sequence of CIM Operations shall be used to boot to the downloaded pre-OS ISO image.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “BootToNetworkISO”
4. Refer to dcim_osdeploy.mof for method input and out parameters and data types.

9.7 Boot to PXE

Following sequence of CIM Operations shall be used to boot to a PXE.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “BootToPXE”

9.8 Detach Drivers

Following sequence of CIM Operations shall be used to detach the USB device containing the drivers from the host server.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “DetachDrivers”

9.9 Download ISO Image to vFlash

Following sequence of CIM Operations shall be used to download a pre-OS ISO image to vFlash.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “DownloadISOToVFlash”
4. Refer to dcim_osdeploy.mof for method input and out parameters and data types.

9.10 Boot to ISO Image from vFlash

Following sequence of CIM Operations shall be used to boot to the pre-OS ISO image that is present in vFlash.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “BootToISOFromVFlash”

4. Refer to dcim_osdeploy.mof for method input and out parameters and data types.

9.11 Detach ISO Image from vFlash

Following sequence of CIM Operations shall be used to detach the ISO Image from vFlash that is attached as a CD-ROM device to host server.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “DetachISOFromVFlash”

9.12 Delete the ISO from vFlash

Following sequence of CIM Operations shall be used to delete the ISO Image from vFlash.

1. Follow the steps (1,2,3) from 9.2
2. Follow the steps (2, 3) from 9.3
3. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “DeleteISOFromVFlash”

9.13 Connect and Attach Network ISO Image

Following sequence of CIM Operations shall be used to connect and attach network ISO Image.

4. Follow the steps (1,2,3) from 9.2
5. Follow the steps (2, 3) from 9.3
6. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “ConnectNetworkISOImage”

9.14 Disconnect and detach Network ISO Image

Following sequence of CIM Operations shall be used to disconnect and detach network ISO Image.

7. Follow the steps (1,2,3) from 9.2
8. Follow the steps (2, 3) from 9.3
9. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “DisconnectNetworkISOImage”

9.15 Get ISO Image connection Status

Following sequence of CIM Operations shall be used to get ISO Image connection status.

10. Follow the steps (1,2,3) from 9.2
11. Follow the steps (2, 3) from 9.3
12. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “GetNetworkISOImageConnectionInfo”

9.16 One time ISO boot skip

Following sequence of CIM Operations shall be used to skip ISO boot once.

13. Follow the steps (1,2,3) from 9.2
14. Follow the steps (2, 3) from 9.3
15. Invoke extrinsic method using the following parameters:
    a. Object path = object path returned from 9.3.3
    b. Method name = “SkipISOImageBoot”

9.17 Status of current Task

Following sequence of CIM Operations shall be used to read the current status of the last method executed.

1. Enumerate (namespace=’root/dcim’, classname=”DCIM_OSDConcreteJob”)
2. Result shall contain one instance of CIM_ConcreteJob
3. The following properties of CIM_ConreteJob shall be used to identify the state and completion status of the last executed method:
    a. InstanceID = “OSD: UnpackAndAttach: 1”
    b. Name = “UnpackAndAttach”
    c. JobStatus = “Completed | Failed”
    d. Refer to dcim_osdeploy.mof for more details on the DCIM_OSDConcreteJob properties.

10 CIM Elements

Table 33 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described; sections 7 “Implementation Requirements” and 8 “Methods” may impose additional requirements on these elements.

Table 33 – CIM Elements: Operating System (OS) Deployment Profile

Element Name	Requirement	Description
Classes		
DCIM_OSDeploymentService	Mandatory	See section 10.1
CIM_ConcreteJob	Conditional	See section 10.2
CIM_RegisteredProfile	Mandatory	See section 10.3
CIM_ElementConformsToProfile	Mandatory	See section 10.4

10.1 DCIM_OSDeploymentService

DCIM_OSDeploymentService is used to provide a central class for the OS Deployment profile.

Table 34 – Class: DCIM_OSDeploymentService

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 OS Deployment Service”.

10.2 CIM_ConcreteJob

CIM_ConcreteJob is used to track the job returned as an OUT parameter of DCIM_OSDeploymentService execute methods. This class is conditional, and shall only exist if one of the extrinsic methods of DCIM_OSDeploymentService returns a job as an output parameter.

Table 35 – Class: CIM_ConcreteJob

Properties and Methods	Requirement	Description
InstanceID	Mandatory	**Key**
Name	Mandatory	
JobState	Optional	
JobStatus	Optional	
OperationalStatus	Mandatory	

10.3 CIM_RegisteredProfile

The CIM_RegisteredProfile class is defined by the Profile Registration Profile. The requirements denoted in the following table are in addition to those mandated by the Profile Registration Profile.

Table 36 – Class: CIM_RegisteredProfile

Properties	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Operating System (OS) Deployment”.

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 “DCIM”

11 Error Message ID and error strings

Here are the list of error message IDs and English error strings

OSD1	The command was successful
OSD2	General failure
OSD3	Lifecycle Controller is being used by another process
OSD4	Cannot access Driver Pack partition in Lifecycle Controller
OSD5	Driver Pack not found in Lifecycle Controller
OSD6	Cannot allocate memory
OSD7	Unable to retrieve Lifecycle Controller handle
OSD8	Setting Boot to PXE through IPMI failed
OSD9	Failed to reboot the system using IPMI command
OSD10	Installation not supported for the selected operating system
OSD11	Driver Pack does not have drivers for the selected operating system
OSD12	Cannot create USB device to copy drivers for the selected operating system
OSD13	Cannot mount USB device to copy drivers for the selected operating system
OSD14	Unable to expose USB device containing operating system drivers to host system
OSD15	Mount network share failed - incorrect username or password
OSD16	Mount network share failed - incorrect IP address or share name
OSD17	Exposing ISO image as internal device to the host system failed
OSD18	Unable to locate the ISO image on the network share point
OSD19	The fork() command for a child process to do the task failed
OSD20	Unable to get size or label from Driver Pack for selected operating system
OSD21	Unable to boot to ISO image
OSD22	Unable to detach ISO image from the host
OSD23	Unable to continue with DetachISOImage - another command is in the process of exposing ISO Image and booting to it
OSD24	Unable to continue with DetachDrivers - UnPackAndAttach is in progress
OSD25	Unable to detach USB device containing operating system drivers
OSD26	Unable to continue with BootToPXE - another command is running
OSD27	Copying drivers for selected operating system failed
OSD28	Hash verification on the ISO image failed
OSD29	Driver Pack config file not found in Lifecycle Controller. Driver Pack might be corrupt
OSD30	Invalid value for ExposeDuration - must be 60 - 65535 seconds
OSD31	Copying operating system drivers to network share failed

OSD32	ISO image is not attached
OSD33	Installed BIOS version does not support this method
OSD34	Unable to continue with BootToPXE - ISO image is attached to the system
OSD35	Lifecycle Controller is disabled
OSD36	Boot to ISO Image has been cancelled by user using CTLR+E option on the server
OSD37	ISO image size too large
OSD38	Copying the ISO image from the network failed
OSD39	Unable to find the VFlash
OSD40	VFlash is not Dell-licensed
OSD41	ISO Image not found on VFlash
OSD42	Downloading ISO File to VFlash failed
OSD43	VFlash unavailable
OSD44	Unable to detach ISO image on VFlash
OSD45	Cannot delete ISO image from VFlash
OSD46	VFlash in use
OSD47	Inaccessible network share
OSD48	ISO Image more than 4GB not supported
OSD49	Disable RIPS mode to access Vflash
OSD50	Lifecycle Controller is in field service mode
OSD51	Reboot the system to run pending System Services Tasks
OSD52	VFlash is Disabled
OSD53	VFlash is write-protected
OSD54	VFlash already has 16 partitions
OSD55	ISO Image is attached to host

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_OSDeploymentService.mof