Operating System (OS) Deployment Profile

Document Number: DCIM1035
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2012-10-25

Version: 1.1.0

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

CONTENTS

Figures

Tables

Introduction

The information in this specification and referenced specifications should be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to represent and manage the OS deployment feature of managed systems and subsystems that are modeled using the DMTF CIM core and extended model definitions.

The target audience for this specification is implementers who are writing CIM-based providers, or consumers of management interfaces that represent the component described in this document.

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: 1.1.0

Organization: DCIM

CIM Schema Version: 2.19.1

Central Class: DCIM_OSDeploymentService

Scoping Class: CIM_ComputerSystem

The 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 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.

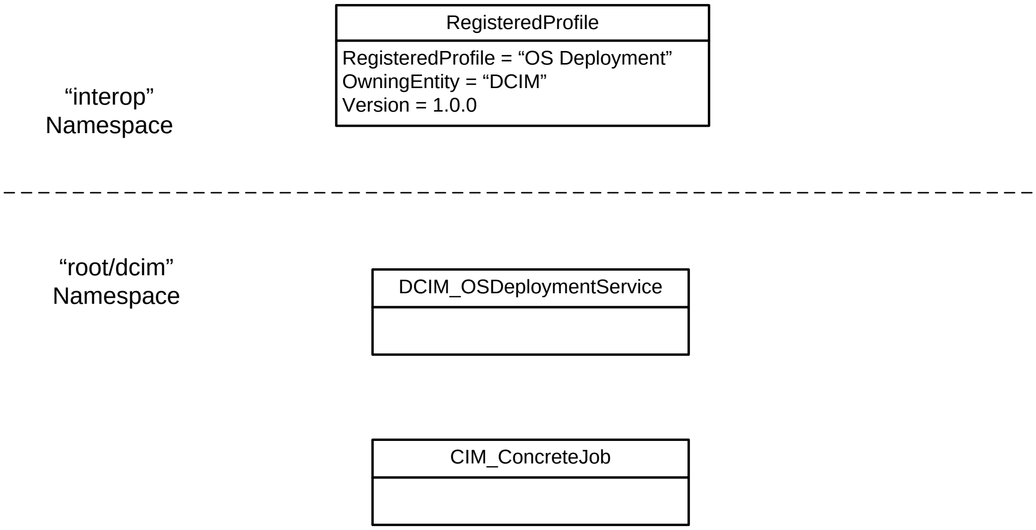

Figure 1 – OS Deployment Profile: Class Diagram

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

		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.

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 – DCIM_OSDeploymentService.GetDriverPackInfo() Method: Parameters.

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.

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 – DCIM_OSDeploymentService.UnpackAndAttach() Method: Parameters.

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.

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 – DCIM_OSDeploymentService.DetachDrivers() Method: Parameters.

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.

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 – DCIM_OSDeploymentService.UnpackAndShare () Method: Parameters.

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	TFTP or NFS share IP address
IN	NFSShare	String	NFS share point
IN	OSName	String	OS name
IN	ShareType	Uint32	0=NFS | 1=TFTP | 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.

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 – DCIM_OSDeploymentService.BootToNetworkISO() Method: Parameters. 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	TFTP or NFS share IP address
IN	NFSShare	String	NFS share point
IN	ImageName	String	ISO image name
IN	ShareType	Uint32	0=NFS | 1=TFTP | 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.

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 – DCIM_OSDeploymentService.DetachISOImage () Method: Parameters.

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.

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 – DCIM_OSDeploymentService.BootToPXE() Method: Parameters. 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.

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 – DCIM_OSDeploymentService.DownloadISOToVFlash() Method: Parameters. 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 IP address
IN	ShareName	String	Network share point
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.

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 – DCIM_OSDeploymentService.BootToISOFromVFlash() Method: Parameters. 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.

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 – DCIM_OSDeploymentService.DetachISOFromVFlash() Method: Parameters.

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.

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 – DCIM_OSDeploymentService.DeleteISOFromVFlash() Method: Parameters.

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.

8.12 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 8.7 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*.

8.13 DCIM_OSDeploymentService Operations

Table 25 lists operations that either have special requirements beyond those from DSP0200, or shall not be supported.

Table 25 – DCIM_OSDeploymentService Operations

Operation	Requirement	Messages
EnumerateInstances	Unspecified	None
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.14 CIM_ConcreteJob

Table 26 lists operations that either have special requirements beyond those from DSP0200, or shall not be supported.

Table 26 – CIM_ConcreteJob Operations

Operation	Requirement	Messages
EnumerateInstances	Unspecified	None
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

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 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.

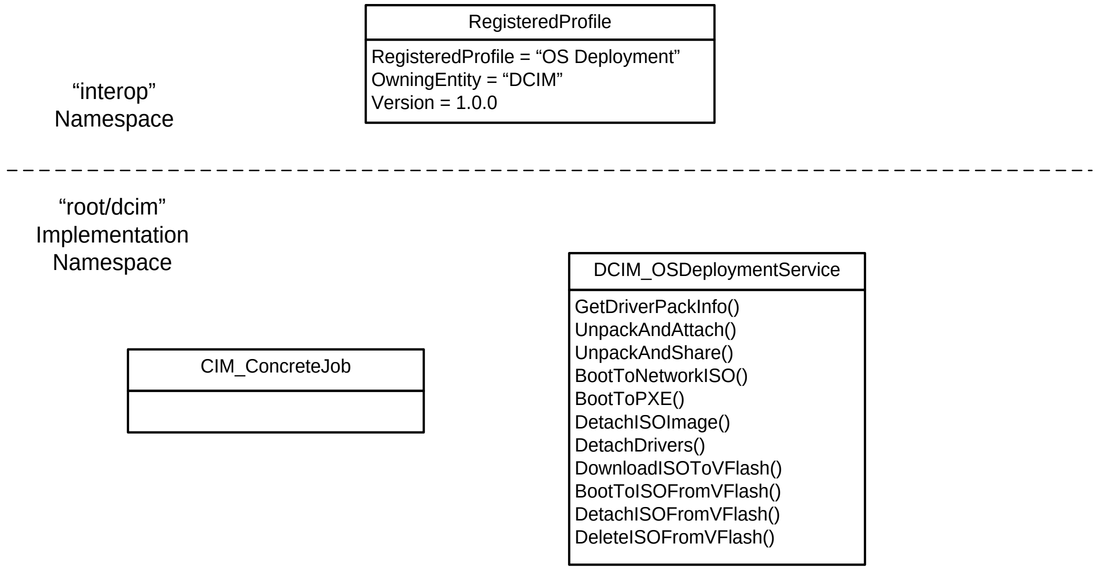

Figure 2 – OS Deployment Profile: Object Diagram

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 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 27 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 27 – CIM Elements: 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 28 – 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 29 – 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 30 – 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”

ANNEX A
(informative)

DCIM Extension MOF

```
 // ==================================================================
// DCIM OS Deployment classes and methods
// ==================================================================
   [ provider ("cmpi:dcimosdeploy"), Description (
       "DCIM_OSDeploymentService is a Dell extension of CIM_Service.
        This class supports several extrinsic methods to support
        remote Operating System deployment.")]
class DCIM_OSDeploymentService : CIM_Service {
      [Description ( "GetDriverPackInfo(). This method returns the list
       of Operating Systems that can be installed on the server using
       embedded device drivers present in the Lifecylce controller." ),
       ValueMap { "0",       "1",              "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 GetDriverPackInfo(
         [IN ( false ), OUT, Description (
             "Version of the driver pack present in the Lifecylce controller"
)]
      string Version,
         [IN ( false ), OUT, Description (
             "List of Operating Systems supported for deployment on the
Server" )]
      string OSList[],
         [IN ( false ), OUT, Description (
             "Reference to a CIM_ConcreteJob" )]
      CIM_ConcreteJob REF Job,
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "UnpackAndAttach(). This method will extract the drivers
for
       the selected Operating System to a USB device that will be attached
locally
       to the server for the specified time interval." ),
       ValueMap { "0",       "1",              "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 UnpackAndAttach(
         [IN, Description (
             "Name of the Operating System to be deployed" )]
      string OSName,
         [IN, Description (
             "duration to expose the drivers" )]
      datetime ExposeDuration,
         [IN ( false ), OUT, Description (
             "Reference to a CIM_ConcreteJob" )]
      CIM_ConcreteJob REF Job,
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "DetachDrivers. This method will detach the USB device
containing
        the drivers from the host server." ),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 DetachDrivers(
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "DetachISOImage. This method will detach the ISO Image
from the host server." ),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 DetachISOImage(
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "BootToPXE. This method will reboot the host server and
boots to PXE." ),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 BootToPXE(
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
```

```
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "UnpackAndShare. This method will extract the drivers
for
       the selected Operating System and copy them to the specified network
share." ),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 UnpackAndShare(
         [IN, Description (
             "Name of the Operating System to be deployed" )]
      string OSName,
         [IN, Description (
             "IP address of the server on which the drivers will be copied"
)]
      string IPAddress,
         [IN, Description (
             "Share name of the server on which the drivers will be copied1"
)]
      string ShareName,
         [IN, Description (
             "Type of the share" ),
             ValueMap { "0",   "1",    "2"},
             Values   { "NFS", "TFTP", "CIFS" }]
      uint16 ShareType,
         [IN, Description (
             "Username of the account to access the share" )]
      string Username,
         [IN, Description (
             "Password of the account to access the share" )]
      string Password,
         [IN, Description (
             "Workgroup of the account to access the share" )]
      string Workgroup,
         [IN ( false ), OUT, Description (
             "Reference to a CIM_ConcreteJob" )]
      CIM_ConcreteJob REF Job,
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "BootToNetworkISO. This method will expose the ISO Image
present on
       network share as a CDROM device to the host server and boots to it."
),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 BootToNetworkISO(
         [IN, Description (
             "IP address of the server that hosts the ISO Image" )]
      string IPAddress,
         [IN, Description (
             "Share name of the server that hosts the ISO Image" )]
      string ShareName,
         [IN, Description (
             "ISO Image name on the server" )]
      string ImageName,
         [IN, Description (
             "Type of the share" ),
             ValueMap { "0",   "2"},
             Values   { "NFS", "CIFS" }]
      uint16 ShareType,
         [IN, Description (
             "Username of the account to access the share" )]
      string Username,
         [IN, Description (
             "Password of the account to access the share" )]
      string Password,
         [IN, Description (
             "Workgroup of the account to access the share" )]
      string Workgroup,
         [IN, Description (
             "Type of Hash algorithm used to compute checksum" ),
             ValueMap { "1",   "2"},
             Values   { "MD5", "SHA1" }]
      uint16 HashType,
         [IN, Description (
             "Checksum value in string format computed using HashType
algorithm" )]
      string HashValue,
         [IN, Description (
             "duration to expose the ISO Image" )]
      datetime ExposeDuration,
         [IN ( false ), OUT, Description (
             "Reference to a CIM_ConcreteJob" )]
      CIM_ConcreteJob REF Job,
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "GetHostMACInfo. This method will return the list of MAC
Addresses
       for all the network devices on the host server." ),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 GetHostMACInfo(
         [IN ( false ), OUT, Description (
             "Returns the list of MAC addressses for all the network devices
on the Host" )]
```

```
      string MACList[],
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "DownloadISOToVFlash. This method will download the ISO
Image present on network share to VFlash." ),
       ValueMap { "0",       "1",             "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 DownloadISOToVFlash(
         [IN, Description (
             "IP address of the server that hosts the ISO Image" )]
      string IPAddress,
         [IN, Description (
             "Share name of the server that hosts the ISO Image" )]
      string ShareName,
         [IN, Description (
             "ISO Image name on the server" )]
      string ImageName,
         [IN, Description (
             "Type of the share" ),
             ValueMap { "0",   "1",    "2"},
             Values   { "NFS", "TFTP", "CIFS" }]
      uint16 ShareType,
         [IN, Description (
             "Username of the account to access the share" )]
      string Username,
         [IN, Description (
             "Password of the account to access the share" )]
      string Password,
         [IN, Description (
             "Workgroup of the account to access the share" )]
      string Workgroup,
         [IN, Description (
             "Type of Hash algorithm used to compute checksum" ),
             ValueMap { "1",   "2"},
             Values   { "MD5", "SHA1" }]
      uint16 HashType,
         [IN, Description (
             "port number to be used" )]
      uint16 Port,
         [IN, Description (
             "Checksum value in string format computed using HashType
algorithm" )]
      string HashValue,
         [IN, Description (
             "duration to expose the ISO Image" )]
      datetime ExposeDuration,
         [IN ( false ), OUT, Description (
             "Reference to a CIM_ConcreteJob" )]
      CIM_ConcreteJob REF Job,
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message 
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "BootToISOFromVFlash. This method will expose the ISO 
Image present on VFlash as a CDROM device to the host server and boots to 
it." ),
       ValueMap { "0",       "1",              "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 BootToISOFromVFlash(
         [IN, Description (
             "duration to expose the ISO Image" )]
      datetime ExposeDuration,
         [IN ( false ), OUT, Description (
             "Reference to a CIM_ConcreteJob" )]
      CIM_ConcreteJob REF Job,
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message 
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "DetachISOFromVFlash. This method will detach the ISO 
Image from the host server." ),
       ValueMap { "0",       "1",              "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 DetachISOFromVFlash(
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message 
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);

      [Description ( "DeleteISOFromVFlash. This method will delete the ISO 
Image present on VFlash." ),
       ValueMap { "0",       "1",              "2",      "4096"},
       Values   { "Success", "Not Supported", "Failed", "Job Created"}]
   uint32 DeleteISOFromVFlash(
         [IN ( false ), OUT, Description (
             "Error message ID part of the OSD message registry" )]
      string MessageID,
         [IN ( false ), OUT, Description (
             "Description of error message in english part of the OSD message
registry" )]
      string Message,
         [IN ( false ), OUT, Description (
             "Substitution variables for the dynamic error messages" )]
      string MessageArguments[]);
};

[ provider("cmpi:dcimosdeploy") ]
class DCIM_OSDConcreteJob : CIM_ConcreteJob
{
   string MessageID;
   string Message;
   string MessageArguments[];
};

[ provider("cmpi:dcimosdeploy") ]
class DCIM_OSDElementConformsToProfile : CIM_ElementConformsToProfile
{
};
```

ANNEX B
(informative)

Items Considered but not Added to Profile

1. OSList *** Needs to be implemented by SW Inventory Provider?

We will do this using the SupportedOS property on the SoftwareIdentity instance that represents the driver package. We will need to differentiate between drivers that may be capable and/or support the following:

- Legacy
- UEFI

Each supported OS will be represented in the SupportedOS array property for each version of Legacy or UEFI.