

Dell™ OS Deployment CIM Profile

Document Number: DELL1035
Document Type: Specification
Document Status: Draft
Document Language: E
Date: 2009-08-28

Version: 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.

© 2008 – 2009 Dell Inc. All rights reserved. Reproduction in any manner whatsoever without the express written permission of Dell, Inc. is strictly forbidden. For more information, contact Dell.

Dell and the *DELL* logo are trademarks of Dell Inc. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

CONTENTS

Foreword	5
Introduction	6
1 Scope	7
2 Normative References.....	7
2.1 Approved References	7
2.2 Other References.....	7
3 Terms and Definitions	7
4 Symbols and Abbreviated Terms	8
5 Synopsis.....	8
6 Description	9
7 Implementation Requirements	10
7.1 DCIM_OSDeploymentService	10
7.2 CIM_ConcreteJob	10
8 Methods.....	12
8.1 Method: DCIM_OSDeploymentService.GetDriverPackInfo()	12
8.2 Method: DCIM_OSDeploymentService.UnpackAndAttach().....	12
8.3 Method: DCIM_OSDeploymentService.DetachDrivers().....	13
8.4 Method: DCIM_OSDeploymentService.UnpackAndShare()	14
8.5 Method: DCIM_OSDeploymentService.BootToNetworkISO()	14
8.6 Method: DCIM_OSDeploymentService.DetachISOImage()	15
8.7 Method: DCIM_OSDeploymentService.BootToPXE()	16
8.8 Profile Conventions for Operations.....	16
8.9 DCIM_OSDeploymentService Operations.....	17
8.10 CIM_ConcreteJob	17
9 Use Cases.....	17
9.1 Object Diagram	18
9.2 Discover OSD profile	18
9.3 Get driver pack version information and supported OSes.....	18
9.4 Unpack drivers and Attach to Host OS.....	19
9.5 Unpack and share.....	19
9.6 Boot to Network ISO image	19
9.7 Boot to PXE	20
9.8 Detach Drivers	20
9.9 Status of current Task.....	20
10 CIM Elements.....	21
10.1 DCIM_OSDeploymentService	21
10.2 CIM_ConcreteJob	21
10.3 CIM_RegisteredProfile.....	22
10.4 CIM_ElementConformToProfile.....	22
ANNEX A Change Log.....	23
ANNEX B DCIM Extension MOF	24

Figures

Figure 1 – OS Deployment Profile: Class Diagram.....	10
Figure 2 – OS Deployment Profile Object Diagram	18

Tables

Table 1 – Related Profiles.....	9
Table 2 – JobStatus	11
Table 3 – DCIM_OSDeploymentService.GetDriverPackInfo() Method: Return Code Values.....	12
Table 4 – DCIM_OSDeploymentService. GetDriverPackInfo() Method: Parameters.....	12
Table 5 – DCIM_OSDeploymentService.UnpackAndAttach() Method: Return Code Values	13
Table 6 – DCIM_OSDeploymentService.UnpackAndAttach() Method: Parameters	13
Table 7 – DCIM_OSDeploymentService.DetachDrivers() Method: Return Code Values	13
Table 8 – DCIM_OSDeploymentService.DetachDrivers() Method: Parameters	13
Table 9 – DCIM_OSDeploymentService. UnpackAndShare() Method: Return Code Values.....	14
Table 10 – DCIM_OSDeploymentService.UnpackAndShare() Method: Parameters.....	14
Table 11 – DCIM_OSDeploymentService.BootToNetworkISO() Method: Return Code Values	15
Table 12 – DCIM_OSDeploymentService.BootToNetworkISO() Method: Parameters.....	15
Table 13 – DCIM_OSDeploymentService. DetachISOImage() Method: Return Code Values	16
Table 14 – DCIM_OSDeploymentService. DetachISOImage() Method: Parameters	16
Table 15 – DCIM_OSDeploymentService. BootToPXE() Method: Return Code Values	16
Table 16 – DCIM_OSDeploymentService. BootToPXE() Method: Parameters	16
Table 17 – DCIM_OSDeploymentService Operations.....	17
Table 18 – CIM_ConcreteJob Operations	17
Table 19 – CIM Elements: OS Deployment Profile.....	21
Table 20 – Class: DCIM_OSDeploymentService	21
Table 21 – Class: CIM_ConcreteJob	21
Table 22 – Class: CIM_RegisteredProfile	22

Foreword

The Dell OS Deployment Profile (DELL1035) was prepared by Dell Engineering. This CIM profile integrates into other CIM profiles published by Dell and the DMTF.

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.

1 OS Deployment Profile

1 Scope

The OS Deployment Profile extends the management capabilities of referencing profiles by adding the capability to represent OS Deployment feature configuration of the Dell Lifecycle Controller. The OS Deployment feature consists of:

Provide supported OS's and version information from the embedded OS driver pack

Unpack the OS Driver Update Package and extract drivers for a specified OS. The drivers are placed on a CIFS or NFS network share.

Unpack the OS Driver Update Package and extract drivers for a specified OS. Expose the unpacked drivers to the host as a USB hard disk device.

Ability to boot to the ISO image present on a network share.

Ability to boot to PXE images.

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

3.1

conditional

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

3.2

mandatory

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

3.3

may

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

3.4

optional

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

3.5

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

3.6

shall

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

3.7

shall not

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

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: OS Deployment

Version: 1.0.0

Organization: Dell

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 Lifecycle Controller. These features include:

Provide supported OS's and version information from the embedded OS driver pack.

Unpack the OS Driver Update Package and extract drivers for a specified OS. The drivers are placed on a CIFS or NFS network share.

Unpack the OS Driver Update Package and extract drivers for a specified OS. Expose the unpacked drivers to the host as a USB hard disk device.

Support booting to a provisioning ISO image located on a network share.

Boot system to PXE.

Get system LOM MAC addresses.

DCIM_OSDeploymentService shall be the Central Class.

Table 1 identifies profiles that are related to this profile.

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration Profile	DMTF	1.0	Mandatory

6 Description

The OS Deployment Profile describes the OS deployment configuration service, and the methods exposed to manage the Lifecycle Controller 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 in the embedded OS driver pack.

Unpack the OS Driver Update Package and extract drivers for a specified OS. The drivers are placed on a CIFS or NFS network share.

Unpack the OS Driver Update Package and extract drivers for a specified OS. Expose the unpacked drivers to the host as a USB hard disk device.

Detach of USB device containing the drivers.

Support booting to a provisioning OS image from a network share.

Detach of ISO image from system.

PXE boot the system.

Get system LOM MAC addresses.

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.

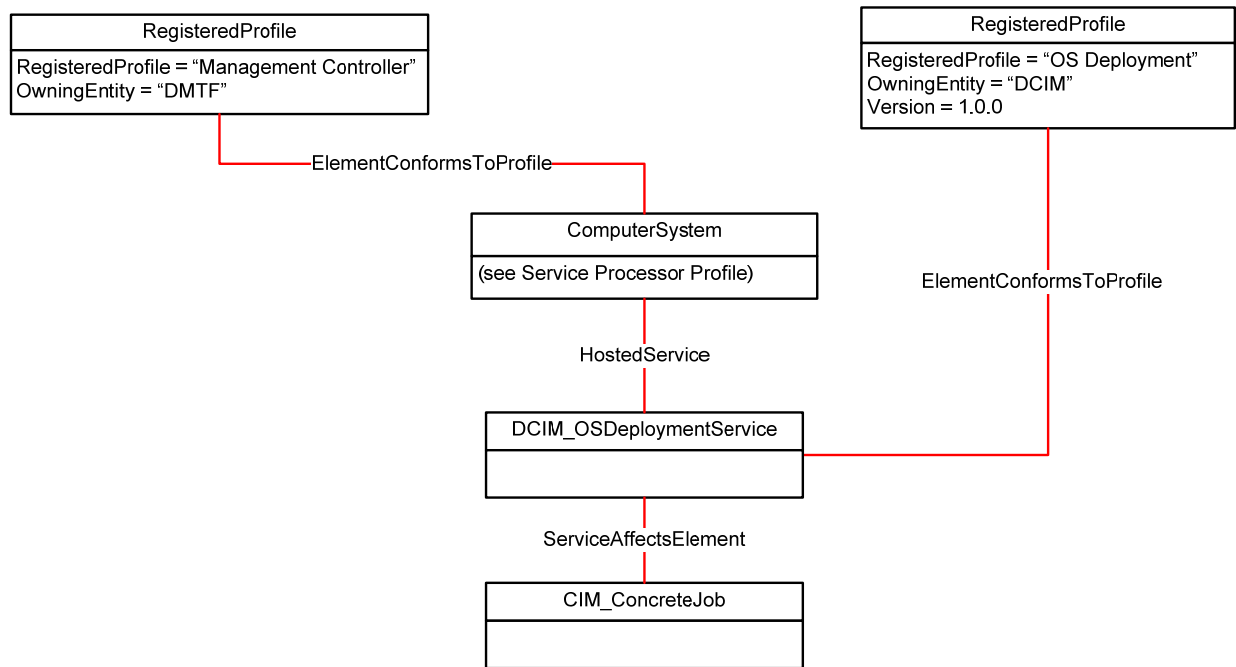


Figure 1 – OS Deployment Profile: Class Diagram

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 JobStatus (free form string) property value shall be one of the entries in Table 2

Table 2 – JobStatus

Job Name	JobStatus	JobStatus Description
"UnpackAndAttach"	Processing Driver Pack	Extracting drivers, creating a 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, please 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, please 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, please 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 operating systems that can be installed on the server using the embedded device drivers present in the Dell Lifecycle Controller product.

Return values for GetDriverPackInfo() shall be as specified in Table 2, where the method-execution behavior matches the return-code description. GetDriverPackInfo() method's parameters are specified in Table 3 – 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 list of operating systems supported for this server.
OUT (optional)	Job	CIM_ConcreteJob	NULL or reference to a CIM_ConcreteJob
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.2 Method: DCIM_OSDeploymentService.UnpackAndAttach()

The UnpackAndAttach() 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.

Return values for UnpackAndAttach() shall be as specified in Table 4, 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 device to be present.
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – USB attach job.
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.3 Method: DCIM_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 4, where the method-execution behavior matches the return-code description. DetachDrivers() method’s parameters are specified in Table 5 – 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	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.4 Method: DCIM_OSDeploymentService.UnpackAndShare()

UnpackAndShare method will extract the drivers for the selected operating system, and copy them to the specified network share.

Return values for UnpackAndShare() shall be as specified in Table 4, where the method-execution behavior matches the return-code description. UnpackAndShare () method's parameters are specified in Table 6 – DCIM_OSDeploymentService.UnpackAndAttach() 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	IP address of TFTP or NFS share
IN	NFSShare	String	NFS share point
IN	OSName	String	Operating System name
IN	ShareType	UInt32	0=NFS 1=TFTP CIFS = 2
IN	Workgroup	String	Workgroup name, if applicable
IN (optional)	UserName	String	Username, if applicable
IN (optional)	Password	String	Password, if applicable
IN (optional)	Port	UInt32	Port num, if applicable
OUT	Job	CIM_ConcreteJob REF	Returned to keep track of OSD – UnpackAndShare job.
OUT (optional)	MessageID	String	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.5 Method: DCIM_OSDeploymentService.BootToNetworkISO()

The BootToNetworkISO() method is used to boot the system to an ISO image located on a CIFS or NFS network share.

Return values for BootToNetworkISO () shall be as specified in Table 9, where the method-execution behavior matches the return-code description. 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	IP address of TFTP or NFS share
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	Username, if applicable
IN (optional)	Password	String	Password, if applicable
IN (optional)	Port	Uint32	Port num, 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	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.6 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 4, where the method-execution behavior matches the return-code description. DetachISOImage () method's parameters are specified in Table 5 – 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	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.7 Method: DCIM_OSDeploymentService.BootToPXE()

The BootToPXE() method is used to boot the server using the PXE mechanism.

Return values for BootToPXE () shall be as specified in Table 10, where the method-execution behavior matches the return-code description. 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	Error MessageID is returned If the method fails to execute.
OUT (optional)	Message	String	Error Message in English corresponding to MessageID is returned If the method fails to execute.

8.8 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 0 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*.

8.9 DCIM_OSDeploymentService Operations

Table 17 lists operations that either have special requirements beyond those from DSP0200, or shall not be supported.

Table 17 – DCIM_OSDeploymentService Operations

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

8.10 CIM_ConcreteJob

Table 18 lists operations that either have special requirements beyond those from DSP0200, or shall not be supported.

Table 18 – 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 the 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.

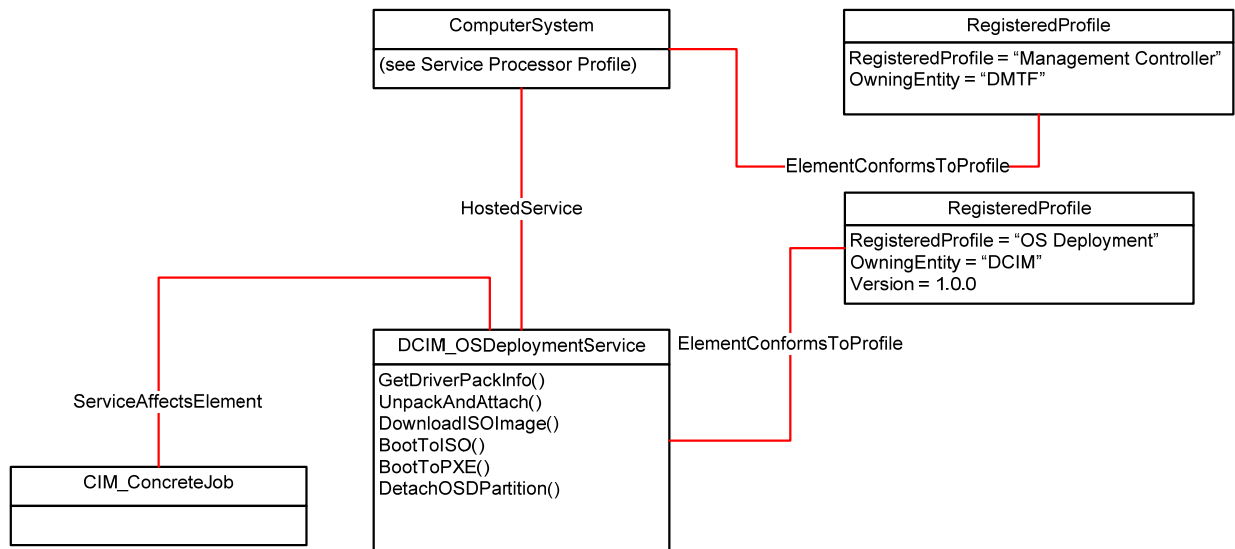


Figure 2 – OS Deployment Profile Object Diagram

9.2 Discover OSD profile

The following sequence of CIM operations shall be used to discover the version of the implemented OSD profile:

1. Enumerate (namespace='root/interop', classname="CIM_RegisteredProfile")
2. Filter the returned enumeration using 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 information and supported OSs

The following sequence of CIM Operations shall be used to retrieve the driver pack version, and supported operating systems, 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_OSDeploymentService") 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

The 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. Please refer to dcim_osdeploy.mof for rest of the method input parameters and their 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

The following sequence of CIM Operations shall be used to extract the drivers for the selected operating system 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. Please refer to dcim_osdeploy.mof for rest of the method input parameters and their data types
4. Please refer to the dcim_osdeploy.mof for rest of the method output parameters.

9.6 Boot to Network ISO image

The following sequence of CIM Operations shall be used to boot to the download 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. Please refer to dcim_osdeploy.mof for method input and out parameters and their data types.

9.7 Boot to PXE

The 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

The 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 Status of current Task

The 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_ConcreteJob 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. Please refer to dcim_osdeploy.mof for more details on the DCIM_OSDConcreteJob properties.

10 CIM Elements

Table 19 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 19Table . Section 7 “Implementation Requirements” and “Methods” may impose additional requirements on these elements.

Table 19 – 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 20 – 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 OUT parameters of DCIM_OSDeploymentService execute methods. This class is conditional, and shall only exist if one of the extrinsic methods of DCIM_OSDeploymentService returns a Jjob as an output parameter.

Table 21 – 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 Table 22 are in addition to those mandated by the Profile Registration Profile.

Table 22 – Class: CIM_RegisteredProfile

Properties	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of "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"

10.4 CIM_ElementConformToProfile

Refer to Profile Registration Profile for more details on this class.

ANNEX A

Change Log

Version	Date	Description
0.0.0	01/20/2009	Initial Draft.
0.1.0	03/03/09	Simplified CIM Model with extrinsic methods
0.3.0	04/09/09	Added 2 new extrinsic methods
0.4.0	08/19/09	Updated based on the latest MOF
1.0.0	08/25/09	Finalized for publication

ANNEX B

DCIM Extension MOF

```
// =====  
// DCIM OS Deployment classes and methods  
// =====  
[ provider ("cmpt: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 Lifecycle 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 Lifecycle  
            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 - can be used to index into Dell  
            Message Registry files" )]  
        string MessageID,  
        [IN ( false ), OUT, Description (  
            "Completed error message in English" )]  
        string Message,  
        [IN ( false ), OUT, Description (  
            "Substitution variables for 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 - can be used to index into Dell
        Message Registry files" )]
string MessageID,
    [IN ( false ), OUT, Description (
        "Completed error message in English" )]
string Message,
    [IN ( false ), OUT, Description (
        "Substitution variables for 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 can be used to index into Dell
        Message Registry files" )]
string MessageID,
    [IN ( false ), OUT, Description (
        "Completed error message in English" )]
string Message,
    [IN ( false ), OUT, Description (
        "Substitution variables for 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 - can be used to index into Dell
        Message Registry files" )]
string MessageID,
    [IN ( false ), OUT, Description (
        "Complete error message in English" )]
string Message,
    [IN ( false ), OUT, Description (
        "Substitution variables for dynamic error messages" )]
string MessageArguments[]);

[Description ( "BootToPXE. This method will reboot the host
    server and boot to PXE." ),
    ValueMap { "0", "1", "2", "4096"},
    Values { "Success", "Not Supported", "Failed", "Job Created"}]
uint32 BootToPXE(
    [IN ( false ), OUT, Description (
        "Error message ID - can be used to index into Dell
        Message Registry files" )]
string MessageID,
    [IN ( false ), OUT, Description (
        "Completed error message in English" )]
string Message,
    [IN ( false ), OUT, Description (
        "Substitution variables for 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 copied" )]
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 - can be used to index into Dell
    Message Registry files" )]
string MessageID,
  [IN ( false ), OUT, Description (
    "Completed error message in English" )]
string Message,
  [IN ( false ), OUT, Description (
    "Substitution variables for dynamic error messages" )]
string MessageArguments[]);

[Description ( "BootToNetworkISO. This method will expose the
  ISO Image present on the specified network share as a CDROM
  device to the host server and boot 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 - can be used to index into Dell
        Message Registry files" )]
string MessageID,
    [IN ( false ), OUT, Description (
        "Completed error message in English" )]
string Message,
    [IN ( false ), OUT, Description (
        "Substitution variables for 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 - can be used to index into Dell
        Message Registry files" )]
string MessageID,
    [IN ( false ), OUT, Description (

```

```

        "Complete error message in English" )]
string Message,
    [IN ( false ), OUT, Description (
        "Substitution variables for dynamic error messages" )]
string MessageArguments[]];
};

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

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

```