Document Number: DSP1070

Date: 2009-10-21

Version: 1.0.1

Opaque Management Data Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

CONTENTS

Foreword 7
Introduction 8
1 Scope 9
2 Normative References 9
2.1 Approved References 9
2.2 Other References 9
3 Terms and Definitions 9
4 Symbols and Abbreviated Terms 11
5 Synopsis 11
6 Description 11
7 Implementation Requirements 13
7.1 CIM_OpaqueManagementDataService 13
7.2 CIM_OpaqueManagementDataCapabilities 14
7.3 Representing Opaque Management Data 16
8 Methods 21
8.1 CIM_OpaqueManagementDataService.Create() 21
8.2 CIM_OpaqueManagementDataService.Read() 22
8.3 CIM_OpaqueManagementDataService.ExportToURI() 23
8.4 CIM_OpaqueManagementDataService.Write() 25
8.5 CIM_OpaqueManagementDataService.ImportFromURI() 26
8.6 CIM_OpaqueManagementDataService.Lock() 28
8.7 CIM_OpaqueManagementDataService.ReassignOwnership() 29
8.8 CIM_OpaqueManagementDataService.AssignAccess() 30
8.9 Profile Conventions for Operations 31
8.10 CIM_OpaqueManagementDataService Operations 31
8.11 CIM_OpaqueManagementData Operations 31
8.12 CIM_OpaqueManagementDataCapabilities Operations 32
8.13 CIM_StorageExtent Operations 32
8.14 CIM_PhysicalMedia Operations 32
8.15 CIM_PhysicalMemory Operations 32
8.16 CIM_HostedService Operations 32
8.17 CIM_ServiceAffectsElement Operations 33
8.18 CIM_ServiceAffectsElementWithQuota Operations 33
8.19 CIM_AssociatedPrivilege Operations 34
8.20 CIM_BasedOn Operations 34
8.21 CIM_ElementCapabilities Operations 34
8.22 CIM_RealizesExtent Operations 35
9 Use Cases 35
9.1 Base Instances 35
9.2 CIM_OpaqueManagementData Instances 36
9.3 Physical Modeling 37
9.4 Create an Instance of CIM_OpaqueManagementData 38
9.5 Read Data Represented by an Instance of CIM_OpaqueManagementData 38
9.6 Write Data Represented by an Instance of CIM_OpaqueManagementData 39
9.7 Overwrite and Truncate Data Represented by an Instance of CIM_OpaqueManagementData 39
9.8 Delete an Instance of CIM_OpaqueManagementData 39
9.9 Enumerate Instances of CIM_OpaqueManagementData 39
9.10 Manage CIM_OpaqueManagementData Access 39
10 CIM Elements 40
10.1 CIM_OpaqueManagementDataService 40

10.2 CIM_OpaqueManagementData ... 41
10.3 CIM_OpaqueManagementDataCapabilities ... 41
10.4 CIM_StorageExtent ... 42
10.5 CIM_PhysicalMedia ... 42
10.6 CIM_PhysicalMemory ... 42
10.7 CIM_RegisteredProfile ... 43
10.8 CIM_HostedService ... 43
10.9 CIM_ServiceAffectsElement (CIM_OpaqueManagementData) ... 43
10.10 CIM_ServiceAffectsElement (CIM_StorageExtent) ... 44
10.11 CIM_ServiceAffectsElement (CIM_Identity) ... 44
10.12 CIM_ServiceAffectsElementWithQuota ... 44
10.13 CIM_AssociatedPrivilege ... 45
10.14 CIM_OpaqueManagementDataOwner ... 45
10.15 CIM_BasedOn ... 45
10.16 CIM_ElementCapabilities ... 45
10.17 CIM_RealizesExtent ... 46
ANNEX A (Informative) Change Log ... 47

Figures

Figure 1 – Opaque Management Data Profile: Profile Class Diagram ... 12
Figure 2 – Base Instances ... 36
Figure 3 – CIM_OpaqueManagementData Instances ... 37
Figure 4 – Physical Modeling ... 38

Tables

Table 1 – Related Profiles ... 11
Table 2 – CIM_OpaqueManagementDataService.Create() Method: Return Code Values ... 21
Table 3 – CIM_OpaqueManagementDataService.Create() Method: Standard Messages ... 21
Table 4 – CIM_OpaqueManagementDataService.Create() Method: Parameters ... 22
Table 5 – CIM_OpaqueManagementDataService.Read() Method: Return Code Values ... 23
Table 6 – CIM_OpaqueManagementDataService.Read() Method: Standard Messages ... 23
Table 7 – CIM_OpaqueManagementDataService.Read() Method: Parameters ... 23
Table 8 – CIM_OpaqueManagementDataService.ExportToURI() Method: Return Code Values ... 24
Table 9 – CIM_OpaqueManagementDataService.ExportToURI() Method: Standard Messages ... 24
Table 10 – CIM_OpaqueManagementDataService.ExportToURI() Method: Parameters ... 24
Table 11 – CIM_OpaqueManagementDataService.Write() Method: Return Code Values ... 25
Table 12 – CIM_OpaqueManagementDataService.Write() Method: Standard Messages ... 25
Table 13 – CIM_OpaqueManagementDataService.Write() Method: Parameters ... 25
Table 14 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Return Code Values ... 26
Table 15 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Standard Messages ... 27
Table 16 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Parameters ... 27
Table 17 – CIM_OpaqueManagementDataService.Lock() Method: Return Code Values ... 28
Table 18 – CIM_OpaqueManagementDataService.Lock() Method: Standard Messages ... 28
Table 19 – CIM_OpaqueManagementDataService.Lock() Method: Parameters ... 28
Table 20 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Return Code Values ... 29

Table 21 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Standard Messages 29
Table 22 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Parameters 29
Table 23 – CIM_OpaqueManagementDataService.AssignAccess() Method: Return Code Values 30
Table 24 – CIM_OpaqueManagementDataService.AssignAccess() Method: Standard Messages........ 30
Table 25 – CIM_OpaqueManagementDataService.AssignAccess() Method: Parameters 31
Table 26 – Operations: CIM_OpaqueManagementData 32
Table 27 – Operations: CIM_HostedService 32
Table 28 – Operations: CIM_ServiceAffectsElement 33
Table 29 – Operations: CIM_ServiceAffectsElementWithQuota 33
Table 30 – Operations: CIM_AssociatedPrivilege 34
Table 31 – Operations: CIM_BasedOn........ 34
Table 32 – Operations: CIM_ElementCapabilities........ 35
Table 33 – Operations: CIM_RealizesExtent........ 35
Table 34 – CIM Elements: Opaque Management Data Profile........ 40
Table 35 – Class: CIM_OpaqueManagementDataService........ 40
Table 36 – Class: CIM_OpaqueManagementData........ 41
Table 37 – Class: CIM_OpaqueManagementDataCapabilities 41
Table 38 – Class: CIM_StorageExtent........ 42
Table 39 – Class: CIM_PhysicalMedia 42
Table 40 – Class: CIM_RegisteredProfile........ 43
Table 41 – Class: CIM_HostedService 43
Table 42 – Class: CIM_ServiceAffectsElement 43
Table 43 – Class: CIM_ServiceAffectsElement 44
Table 44 – Class: CIM_ServiceAffectsElement 44
Table 45 – Class: CIM_ServiceAffectsElementWithQuota 44
Table 46 – Class: CIM_AssociatedPrivilege........ 45
Table 47 – Class: CIM_BasedOn 45
Table 48 – Class: CIM_ElementCapabilities........ 45
Table 49 – Class: CIM_RealizesExtent 46

Foreword

The *Opaque Management Data Profile* (DSP1070) was prepared by the Physical Platform Profiles Working Group of the DMTF.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Acknowledgments

The authors wish to acknowledge the following people.

Editor:

- David Hines – Intel

Contributors:

- Murali Rajagopal
- Hemal Shah – Broadcom
- Sharon L Smith – Intel

Participants from the DMTF Desktop and Mobile Working Group:

- Bob Blair – AMD
- Jon Hass – Dell
- Jeff Hilland – HP
- Joel Clark – Intel
- Steve Hand – Symantec

Introduction

The information in this specification 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 opaque management data that is modeled using the DMTF Common Information Model (CIM) core and extended model definitions.

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

Opaque Management Data Profile

1 Scope

The *Opaque Management Data Profile* extends the management capability of referencing profiles by adding the capability to create, manage, use, and delete opaque management data objects.

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 DSP0004, *CIM Infrastructure Specification 2.5*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

DMTF DSP0200, *CIM Operations over HTTP 1.2,*
http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.0,*
http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

DMTF DSP1011, *Physical Asset Profile 1.0,*
http://www.dmtf.org/standards/published_documents/DSP1011_1.0.pdf

DMTF DSP1033, *Profile Registration Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

DMTF DSP1034, *Simple Identity Management Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1034_1.0.pdf

DMTF DSP1058, *Base Desktop and Mobile Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1058_1.0.pdf

DMTF DSP1080, *Enabled Logical Element Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1080_1.0.pdf

DMTF DSP8016, *WBEM Operations Registry,*
http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016.xml

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

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. For the purposes of this document, the terms and definitions given in DSP1033 and DSP1001 also 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 in order to conform to the document and from which no deviation is permitted

3.11

should

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

3.12

should not

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

3.13

unspecified

indicates that this profile does not define any constraints for the referenced CIM element or operation

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1

CIM

Common Information Model

5 Synopsis

Profile Name: Opaque Management Data

Version: 1.0.1

Organization: DMTF

CIM Schema Version: 2.22

Central Class: CIM_OpaqueManagementDataService

Scoping Class: CIM_ComputerSystem

The *Opaque Management Data Profile* is a component profile that provides the ability to manage and use opaque management data on a system.

CIM_OpaqueManagementDataService shall be the Central Class of this profile. The instance(s) of CIM_OpaqueManagementDataService shall be the Central Instance(s) of this profile. CIM_ComputerSystem shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem with which the Central Instance is associated through an instance of CIM_HostedService shall be the Scoping Instance of this profile.

Table 1 identifies profiles on which this profile has a dependency.

Table 1 – Related Profiles

Profile Name	Organization	Version	Requirement
Profile Registration	DMTF	1.0	Mandatory
Simple Identity Management	DMTF	1.0	Mandatory
Physical Asset	DMTF	1.0	Optional
Enabled Logical Element	DMTF	1.0	Mandatory

6 Description

The *Opaque Management Data Profile* defines capabilities for producing, consuming, and describing “opaque” management data in managed systems. The structure and semantics of opaque management data are known to management client software, but may not be known to the implementation. Opaque data are stored in non-volatile memory that is accessible to the implementation. Management clients are able to read and write opaque data, as well as manage its lifetime and access rights. The non-volatile storage is often used as a “mailbox”, with local management components producing data for consumption by remote management clients, or vice versa. The information in this specification should be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that must be instantiated and manipulated to represent and manage the opaque management data of systems and subsystems modeled using the DMTF CIM core and extended model definitions.

Figure 1 represents the class schema for the *Opaque Management Data Profile*. For simplicity, the prefix CIM_ has been removed from the names of the classes.

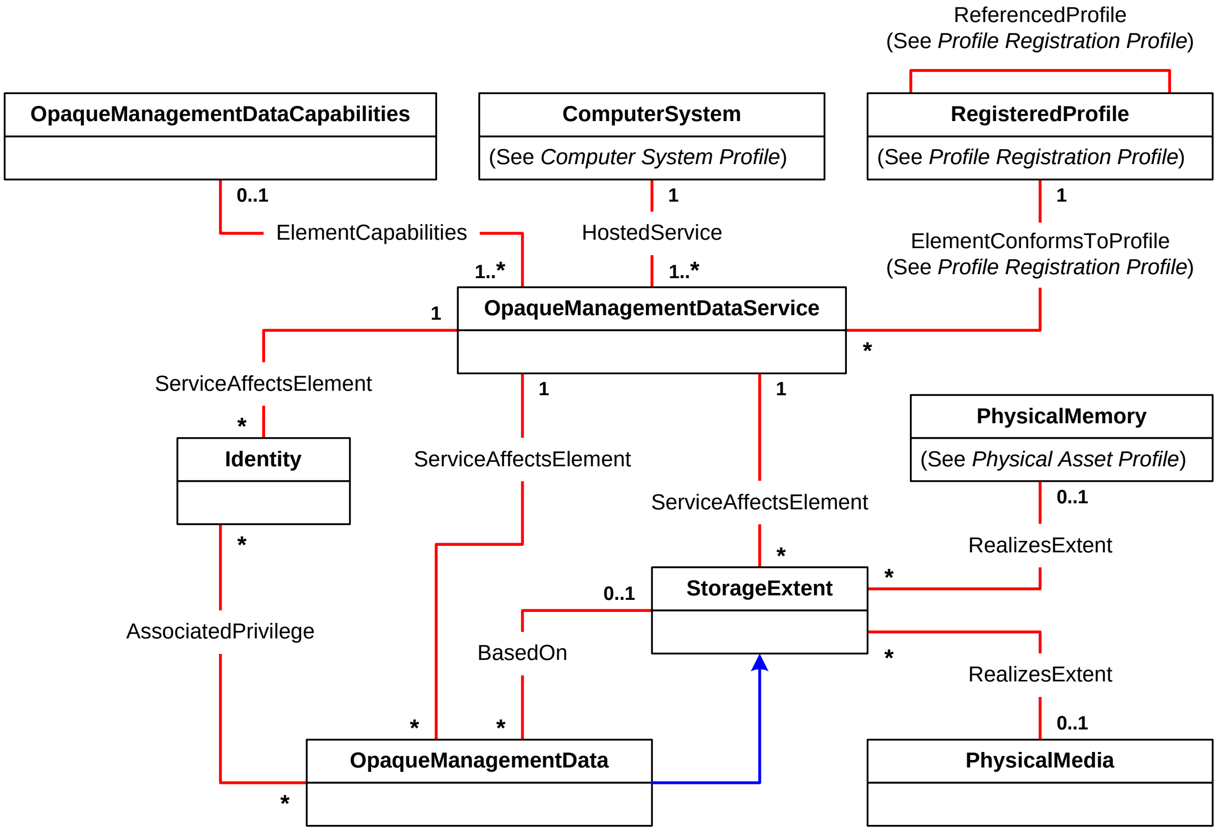

Figure 1 – Opaque Management Data Profile: Profile Class Diagram

The central class is CIM_OpaqueManagementDataService, which supports extrinsic methods for creating, reading, writing, and locking opaque management data. CIM_OpaqueManagementDataService is associated to the scoping class through CIM_HostedService.

CIM_OpaqueManagementDataService capabilities are described by CIM_OpaqueManagementDataCapabilities, a specialization of CIM_EnabledLogicalElementCapabilities. CIM_OpaqueManagementDataCapabilities is associated with CIM_OpaqueManagementDataService through CIM_ElementCapabilities.

Logical units of opaque management data are represented by CIM_OpaqueManagementData, which contains properties that describe, but do not access the opaque management data. All access to opaque management data itself is done through CIM_OpaqueManagementDataService extrinsic methods. CIM_OpaqueManagementData is a specialization of CIM_StorageExtent.

CIM_OpaqueManagementData is managed by CIM_OpaqueManagementDataService. This relationship is represented through the association class CIM_ServiceAffectsElement.

CIM_OpaqueManagementData may be based on a larger CIM_StorageExtent. This relationship is represented through the association class CIM_BasedOn. The impact of storing opaque management data on this larger extent is in turn represented through the association class CIM_ServiceAffectsElement between CIM_StorageExtent and CIM_OpaqueManagementDataService.

Owners and users of CIM_OpaqueManagementData are represented by CIM_Identity. The relationship between CIM_OpaqueManagementData and its users and owners is represented by CIM_AssociatedPrivilege and its subclass CIM_OpaqueManagementDataOwner, respectively.

The CIM_OpaqueManagementDataService may impose limits on the amount of storage users may allocate for opaque management data by assigning a quota, represented by CIM_ServiceAffectsElementWithQuota.

The physical aspects of the storage used for opaque management data are represented by CIM_PhysicalMemory or CIM_PhysicalMedia, depending on the type of physical storage used. These are associated to CIM_StorageExtent through CIM_RealizesExtent.

Support for the *Opaque Management Data Profile* is advertised by CIM_RegisteredProfile.

7 Implementation Requirements

This clause specifies the requirements related to the arrangement of instances and properties for implementations of this profile. Methods are specified in clause 8, and properties are specified in clause 10.

7.1 CIM_OpaqueManagementDataService

At least one instance of CIM_OpaqueManagementDataService shall be supported.

7.1.1 CIM_OpaqueManagementDataService.LockTimeout

The CIM_OpaqueManagementDataService extrinsic method Lock() allows a client to lock an instance of CIM_OpaqueManagementData.

An implementation that supports locking CIM_OpaqueManagementData may also support an automatic lock timeout. A lock timeout is a duration that represents the maximum amount of time that a lock on an instance of CIM_OpaqueManagementData shall remain in effect without any access of the data represented by the CIM_OpaqueManagementData object by the Identity that holds the lock.

If an implementation supports locking CIM_OpaqueManagementData and supports an automatic lock timeout, then CIM_OpaqueManagementDataService.LockTimeout shall be supported and shall contain the duration of this timeout.

If an implementation does not support locking CIM_OpaqueManagementData or does not support an automatic lock timeout, then CIM_OpaqueManagementDataService.LockTimeout shall be NULL.

7.1.2 CIM_OpaqueManagementDataService.MaxCreateSize

CIM_OpaqueManagementDataService.MaxCreateSize may be supported. If supported, CIM_OpaqueManagementDataService.MaxCreateSize shall contain the largest value that the implementation will currently accept in the MaxSize parameter in a call to CIM_OpaqueManagementDataService.Create().

7.1.3 CIM_OpaqueManagementDataService.AvailableStorage

CIM_OpaqueManagementDataService.AvailableStorage shall contain the total number of free bytes of storage currently available for allocation to CIM_OpaqueManagementData instances. Accordingly, the number of bytes of storage already allocated to CIM_OpaqueManagementData instances shall be equal to the difference between CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage and CIM_OpaqueManagementDataService.AvailableStorage.

CIM_OpaqueManagementDataService.AvailableStorage shall not exceed CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage.

7.1.4 Managing CIM_OpaqueManagementDataService.ElementName

An implementation may support management of the ElementName property of CIM_OpaqueManagementDataService. The abstract *Enabled Logical Element Profile* specifies requirements for supporting ElementName management in subclasses of CIM_EnabledLogicalElement. The implementation of CIM_OpaqueManagementDataService shall meet the requirements in the *Enabled Logical Element Profile*, with CIM_OpaqueManagementDataService in place of CIM_EnabledLogicalElement and CIM_OpaqueManagementDataCapabilities in place of CIM_EnabledLogicalElementCapabilities.

7.1.5 Managing CIM_OpaqueManagementDataService State

An implementation may support management of CIM_OpaqueManagementDataService state. The abstract *Enabled Logical Element Profile* specifies requirements for supporting state management in subclasses of CIM_EnabledLogicalElement. The implementation of CIM_OpaqueManagementDataService shall meet the requirements of the *Enabled Logical Element Profile*, with CIM_OpaqueManagementDataService in place of CIM_EnabledLogicalElement and CIM_OpaqueManagementDataCapabilities in place of CIM_EnabledLogicalElementCapabilities.

7.1.6 CIM_HostedService

Each instance of CIM_OpaqueManagementDataService shall be associated with the scoping instance of CIM_ComputerSystem through an instance of CIM_HostedService.

7.2 CIM_OpaqueManagementDataCapabilities

Exactly one instance of CIM_OpaqueManagementDataCapabilities shall be associated with each instance of CIM_OpaqueManagementDataService through the CIM_ElementCapabilities association.

7.2.1 CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage

CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage shall contain the total number of bytes of storage for opaque management data managed by the instance of CIM_OpaqueManagementDataService associated with the CIM_OpaqueManagementDataCapabilities instance through CIM_ElementCapabilities. This number shall include all allocated and unallocated storage.

7.2.2 CIM_OpaqueManagementDataCapabilities.MaxReadLength

CIM_OpaqueManagementDataCapabilities.MaxReadLength shall contain the maximum value that the implementation supports in the Length parameter of the CIM_OpaqueManagementDataService.Read() and CIM_OpaqueManagementDataService.ExportToURI() methods. The value of CIM_OpaqueManagementDataCapabilities.MaxReadLength may be less than the value in the DataSize property of a CIM_OpaqueManagementData instance. A client that wishes to read more bytes than the value of CIM_OpaqueManagementDataCapabilities.MaxReadLength should do so by using CIM_OpaqueManagementDataService.Read() multiple times.

7.2.3 CIM_OpaqueManagementDataCapabilities.MaxWriteLength

CIM_OpaqueManagementDataCapabilities.MaxWriteLength shall contain the maximum value that the implementation supports in the Length parameter of the CIM_OpaqueManagementDataService.Write() and and CIM_OpaqueManagementDataService.ImportFromURI() methods. The value of CIM_OpaqueManagementDataCapabilities.MaxWriteLength may be less than the value in the MaxSize property of a CIM_OpaqueManagementData instance. A client that wishes to write more bytes than the value of CIM_OpaqueManagementDataCapabilities.MaxWriteLength should do so by using CIM_OpaqueManagementDataService.Write() multiple times.

7.2.4 CIM_OpaqueManagementDataCapabilities.SupportedMethods

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain each value that corresponds to a CIM_OpaqueManagementDataService method supported by the implementation, and shall not contain any value that corresponds to a CIM_OpaqueManagementDataService method not supported by the implementation.

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 2 (Create) if and only if the implementation supports the method CIM_OpaqueManagementDataService.Create().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 3 (Read) if and only if the implementation supports the method CIM_OpaqueManagementDataService.Read().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 4 (ExportToURI) if and only if the implementation supports the method CIM_OpaqueManagementDataService.ExportToURI().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 5 (Write) if and only if the implementation supports the method CIM_OpaqueManagementDataService.Write().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 6 (ImportFromURI) if and only if the implementation supports the method CIM_OpaqueManagementDataService.ImportFromURI().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 7 (Lock) if and only if the implementation supports the method CIM_OpaqueManagementDataService.Lock().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 8 (ReassignOwnership) if and only if the implementation supports the method CIM_OpaqueManagementDataService.ReassignOwnership().

CIM_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 9 (AssignAccess) if and only if the implementation supports the method CIM_OpaqueManagementDataService.AssignAccess().

7.2.5 CIM_OpaqueManagementDataCapabilities.SupportedExportURISchemes

The CIM_OpaqueManagementDataCapabilities.SupportedExportURISchemes property shall list the URI schemes that are supported by the implementation for. CIM_OpaqueManagementDataService.ExportToURI().

7.2.6 CIM_OpaqueManagementDataCapabilities.SupportedImportURISchemes

The CIM_OpaqueManagementDataCapabilities.SupportedImportURISchemes property shall list the URI schemes that are supported by the implementation for. CIM_OpaqueManagementDataService.ImportFromURI().

7.3 Representing Opaque Management Data

Zero or more instances of CIM_OpaqueManagementData shall be instantiated. Each CIM_OpaqueManagementData instance shall represent an independent logical unit of opaque management data.

7.3.1 Managing CIM_OpaqueManagementData.ElementName

An implementation may support management of the ElementName property of CIM_OpaqueManagementData. The abstract *Enabled Logical Element Profile* specifies requirements for supporting ElementName management in subclasses of CIM_EnabledLogicalElement. The implementation of CIM_OpaqueManagementData shall meet the requirements in the *Enabled Logical Element Profile*, with CIM_OpaqueManagementData in place of CIM_EnabledLogicalElement.

7.3.2 Managing CIM_OpaqueManagementData State

An implementation may support management of CIM_OpaqueManagementData state. The abstract *Enabled Logical Element Profile* specifies requirements for supporting state management in subclasses of CIM_EnabledLogicalElement. The implementation of CIM_OpaqueManagementData shall meet the requirements of the *Enabled Logical Element Profile*, with CIM_OpaqueManagementData in place of CIM_EnabledLogicalElement.

7.3.3 Ownership of Opaque Management Data

Each instance of CIM_OpaqueManagementData shall be owned by exactly one user, represented by an instance of CIM_Identity. The owner of each CIM_OpaqueManagementData instance created by a client using CIM_OpaqueManagementDataService.Create() shall be the CIM_Identity instance referenced by the Owner parameter in that method.

7.3.3.1 CIM_OpaqueManagementDataOwner

Every instance of CIM_OpaqueManagementData shall have exactly one owner. Exactly one instance of CIM_OpaqueManagementDataOwner shall exist between each instance of CIM_OpaqueManagementData and the CIM_Identity instance that owns it. The requirements defined in clause 7.3.4.1 for instances of CIM_AssociatedPrivilege shall also be met for instances of its subclass CIM_OpaqueManagementDataOwner.

7.3.3.2 Reassignment of Ownership

An implementation may support reassignment of ownership of a CIM_OpaqueManagementData instance by its owner CIM_Identity instance to another CIM_Identity instance.

7.3.3.2.1 Reassignment of Ownership Is Supported

If an implementation supports reassignment of ownership of a CIM_OpaqueManagementData instance by its owner CIM_Identity instance to another CIM_Identity instance, it shall meet the requirements specified in this clause.

7.3.3.2.1.1 CIM_OpaqueManagementDataService.ReassignOwnership()

The CIM_OpaqueManagementDataService.ReassignOwnership() method shall be implemented and shall not return 1 (Not supported).

7.3.3.2.1.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods

The CIM_OpaqueManagementDataCapabilities.SupportedMethods array property shall contain the value 8 (ReassignOwnership).

7.3.3.2.2 Reassignment of Ownership Is Not Supported

If an implementation does not support reassignment of ownership of a CIM_OpaqueManagementData instance by its owner CIM_Identity instance to another CIM_Identity instance, it shall meet the requirements specified in this clause.

7.3.3.2.2.1 CIM_OpaqueManagementDataService.ReassignOwnership()

The CIM_OpaqueManagementDataService.ReassignOwnership() method shall not be implemented.

7.3.3.2.2.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods

The CIM_OpaqueManagementDataCapabilities.SupportedMethods array property shall not contain the value 8 (ReassignOwnership).

7.3.4 Access Rights to Opaque Management Data

An implementation shall support read and write access to opaque management data, where “reading” corresponds to executing CIM_OpaqueManagementDataService.Read() or CIM_OpaqueManagementDataService.ExportToURI() and “writing” corresponds to executing CIM_OpaqueManagementDataService.Write() or CIM_OpaqueManagementDataService.ImportOpaqueManagementDataToURI().

7.3.4.1 CIM_AssociatedPrivilege

An instance of CIM_AssociatedPrivilege or its subclass CIM_OpaqueManagementDataOwner shall exist between each CIM_OpaqueManagementData instance and each CIM_Identity instance that has access to it. If the CIM_Identity instance is the owner of the CIM_OpaqueManagementData instance, this shall be an instance of CIM_OpaqueManagementDataOwner. If the CIM_Identity instance is not the owner of the CIM_OpaqueManagementData instance, this shall be an instance of CIM_AssociatedPrivilege.

7.3.4.1.1 CIM_AssociatedPrivilege.Activities

Read access and write access shall be represented by the values 5 (Read), and 6 (Write), respectively, in CIM_AssociatedPrivilege.Activities.

The Activities property shall contain the value 5 (Read) if and only if the CIM_Identity instance is permitted to execute CIM_OpaqueManagementDataService.Read() and CIM_OpaqueManagementDataService.ExportToURI() on the associated CIM_OpaqueManagementData. The Activities property shall not contain the value 5 (Read) in more than one element.

The Activities property shall contain the value 6 (Write) if and only if the CIM_Identity instance is permitted to execute CIM_OpaqueManagementDataService.Write() and CIM_OpaqueManagementDataService.ImportFromURI() on the associated CIM_OpaqueManagementData instance. The Activities property shall not contain the value 6 (Write) in more than one element.

The Activities property shall not contain any value other than 5 (Read) or 6 (Write).

7.3.4.1.2 CIM_AssociatedPrivilege.ActivityQualifiers

The value “DMTF:DSP1070:” shall appear in each entry that corresponds to an entry in CIM_AssociatedPrivilege.Activities that conforms to clause 7.3.4.1.1.

7.3.4.1.3 CIM_AssociatedPrivilege.QualifierFormats

The value 12 (Specification-defined) shall appear in each entry that corresponds to an entry in CIM_AssociatedPrivilege.Activities that conforms to clause 7.3.4.1.1.

7.3.4.2 Assignment of Access Rights

An implementation may support assignment and removal of read and write access rights on a CIM_OpaqueManagementData instance by its owner CIM_Identity instance to any CIM_Identity instances, including itself.

7.3.4.2.1 Assignment of Access Rights Is Supported

If an implementation supports assignment of read and write access rights on a CIM_OpaqueManagementData instance by its owner CIM_Identity instance to other CIM_Identity instances, it shall meet the requirements specified in this clause.

7.3.4.2.1.1 CIM_OpaqueManagementDataService.AssignAccess()

The CIM_OpaqueManagementDataService.AssignAccess() method shall be implemented and shall not return 1 (Not supported).

7.3.4.2.1.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods

The CIM_OpaqueManagementDataCapabilities.SupportedMethods array property shall contain the value 9 (AssignAccess).

7.3.4.2.2 Assignment of Access Rights Is Not Supported

If an implementation does not support assignment of read and write access rights on a CIM_OpaqueManagementData instance by its owner CIM_Identity instance to other CIM_Identity instances, it shall meet the requirements specified in this clause.

7.3.4.2.2.1 CIM_OpaqueManagementDataService.AssignAccess()

The CIM_OpaqueManagementDataService.AssignAccess() method shall not be implemented.

7.3.4.2.2.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods

The CIM_OpaqueManagementDataCapabilities.SupportedMethods array property shall not contain the value 9 (AssignAccess).

7.3.5 CIM_OpaqueManagementData.DataSize

CIM_OpaqueManagementData.DataSize shall contain the number of logically contiguous bytes of meaningful data in the CIM_OpaqueManagementData instance, starting at offset 0, and ending at offset DataSize – 1.

The initial value of DataSize upon creation of the CIM_OpaqueManagementData instance and before any writes shall be 0.

The value of DataSize shall not be greater than the value in CIM_OpaqueManagementData.MaxSize.

An implementation shall support logical truncation of CIM_OpaqueManagementData objects through reduction of the value of the DataSize property. Such a reduction shall logically eliminate any data bytes at offsets greater than or equal to the new value of the DataSize property.

An implementation shall fail an attempt to increase the value of the DataSize property.

7.3.6 CIM_OpaqueManagementData.MaxSize

CIM_OpaqueManagementData.MaxSize shall contain the maximum number of total bytes that can be written into an instance of CIM_OpaqueManagementData.

7.3.7 CIM_OpaqueManagementData.DataFormat

CIM_OpaqueManagementData.DataFormat shall contain a string that describes the format of the data in an instance of CIM_OpaqueManagementData.

7.3.8 CIM_OpaqueManagementData.LastAccessed

If an implementation supports the CIM_OpaqueManagementData.LastAccessed property, it shall contain the last date and time the data represented by the instance of CIM_OpaqueManagementData was accessed by a CIM_OpaqueManagementDataService extrinsic method or non-CIM access mechanism.

7.3.9 CIM_OpaqueManagementData.WriteLimited

If the storage extent upon which the CIM_OpaqueManagementData is based has a maximum number of write or erase operations associated with it, CIM_OpaqueManagementData.WriteLimited shall contain 2 (Write-limited).

If the storage extent upon which the CIM_OpaqueManagementData is based does not have a maximum number of write or erase operations associated with it, CIM_OpaqueManagementData.WriteLimited shall contain 1 (Not write-limited).

If it is unknown whether the storage extent upon which the CIM_OpaqueManagementData is based has a maximum number of write or erase operations associated with it, CIM_OpaqueManagementData.WriteLimited shall contain 0 (Unknown).

7.3.10 Representing the Storage of Opaque Management Data

An implementation may support the use of CIM_StorageExtent to describe the storage in which CIM_OpaqueManagementData instances are stored. If an implementation does support CIM_StorageExtent for this purpose, each CIM_OpaqueManagementData instance shall be associated with an instance of CIM_StorageExtent through an instance of CIM_BasedOn.

7.3.11 Representing the Physical Realization of Opaque Management Data

An implementation may support the use of CIM_PhysicalMedia or CIM_PhysicalMemory to describe the physical realization of CIM_OpaqueManagementData instances or the CIM_StorageExtent upon which there are based. If an implementation does support CIM_PhysicalMedia or CIM_PhysicalMemory for this purpose, each CIM_StorageExtent instance shall be associated with an instance of CIM_PhysicalMedia or CIM_PhysicalMemory through an instance of CIM_Realizes.

7.3.12 Representing Management of Opaque Management Data

Each instance of CIM_OpaqueManagementData shall be associated with exactly one instance of CIM_OpaqueManagementDataService through CIM_ServiceAffectsElement or its subclass CIM_ServiceAffectsElementWithQuota.

Each instance of CIM_StorageExtent that is associated with one or more instances of CIM_OpaqueManagementData through CIM_BasedOn shall be associated through an instance of CIM_ServiceAffectsElement with the instance of CIM_OpaqueManagementDataService that is associated with the CIM_OpaqueManagementData instance through CIM_ServiceAffectsElement or its subclass CIM_ServiceAffectsElementWithQuota.

7.3.13 Representing Management of Users of Opaque Management Data

The ServiceAffectsElement association shall be used to model the identities permitted to access opaque management data.

7.3.13.1 CIM_ServiceAffectsElement

Each instance of CIM_Identity that represents a user of opaque management data shall be connected through an instance of CIM_ServiceAffectsElement to each instance of CIM_OpaqueManagementDataService that is associated to at least one instance of CIM_OpaqueManagementData that is used by the user.

7.3.13.2 Representing Opaque Management Data Storage Quotas

An implementation may support assignment of opaque management data storage quotas to CIM_Identity instances. The CIM_ServiceAffectsElement specialization CIM_ServiceAffectsElementWithQuota shall be used to represent such quotas. If an implementation supports assignment of opaque management data storage quotas to CIM_Identity instances, the requirements in this clause shall be met.

7.3.13.2.1 CIM_ServiceAffectsElementWithQuota

The creation class of the CIM_ServiceAffectsElement instance between the CIM_Identity instance and the CIM_OpaqueManagementDataService instance that assigns the quota shall be CIM_ServiceAffectsElementWithQuota.

7.3.13.2.1.1 CIM_ServiceAffectsElementWithQuota.Quota

CIM_ServiceAffectsElementWithQuota.Quota shall contain the quota value, in bytes, that has been assigned to the CIM_Identity instance associated through CIM_ServiceAffectsElementWithQuota to the instance of CIM_OpaqueManagementDataService.

7.3.13.2.1.2 CIM_ServiceAffectsElementWithQuota.QuotaUsed

CIM_ServiceAffectsElementWithQuota.QuotaUsed shall contain the number of bytes of the quota value that have been used by the CIM_Identity instance associated through CIM_ServiceAffectsElementWithQuota to the instance of CIM_OpaqueManagementDataService.

7.3.13.2.1.3 CIM_ServiceAffectsElementWithQuota.QuotaUnits

CIM_ServiceAffectsElementWithQuota.QuotaUnits shall contain “byte”.

7.3.14 Locking Opaque Management Data

An implementation may support concurrent access to CIM_OpaqueManagementData instances by multiple clients. If the CIM_OpaqueManagementDataService.Read() and CIM_OpaqueManagementDataService.Write() operations of multiple clients on a single CIM_OpaqueManagementData instance are interlaced, data incoherence or corruption may result.

To avoid data incoherence and corruption, an implementation may support a lock on each CIM_OpaqueManagementData instance.

7.3.14.1 Locking Opaque Management Data Is Supported

The requirements in this clause shall be met by an implementation that supports locking CIM_OpaqueManagementData instances.

The CIM_OpaqueManagementDataService method Lock() shall be implemented and shall not return 1 (Not supported).

7.3.14.2 Locking Opaque Management Data Is Supported

The requirements in this clause shall be met by an implementation that does not support locking CIM_OpaqueManagementData instances.

The CIM_OpaqueManagementDataService method Lock() shall not be implemented, or shall be implemented and always return 1 (Not supported).

8 Methods

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

8.1 CIM_OpaqueManagementDataService.Create()

If an implementation supports creation of CIM_OpaqueManagementData instances by clients, it shall support the CIM_OpaqueManagementDataService.Create() method. The client that invokes this method shall be the initial owner of the CIM_OpaqueManagementData instance that it creates.

If the CIM_OpaqueManagementDataService.Create() method succeeds, it shall create the following instances:

1) A single CIM_OpaqueManagementData instance.

2) A single CIM_OpaqueManagementDataOwner instance between the new CIM_OpaqueManagementData instance and the CIM_Identity instance referenced by the Owner parameter of the method. The Activities properties of the new CIM_OpaqueManagementDataOwner instance shall contain the values 5 (Read) and 6 (Write).

The return code values for the CIM_OpaqueManagementDataService.Create() method are specified in Table 2.

Table 2 – CIM_OpaqueManagementDataService.Create() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred
4096	Job started

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.Create() method are specified in Table 3.

Table 3 – CIM_OpaqueManagementDataService.Create() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found
WIP219	CIM method not supported by CIM class implementation

The parameters for the CIM_OpaqueManagementDataService.Create() method are specified in Table 4.

Table 4 – CIM_OpaqueManagementDataService.Create() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	MaxSize	uint64	The value of the MaxSize parameter shall be used as the basis for the value of the MaxSize property of the CIM_OpaqueManagementData instance to be created by this method. If the value of MaxSize is greater than the value of the MaxCreateSize property in the central instance then the method shall fail. If the method succeeds, the value of the MaxSize property in the newly created instance shall not be less than the value of this parameter.
IN	DataFormat	string	The value of the DataFormat parameter shall be copied into the DataFormat property of the CIM_OpaqueManagementData instance to be created by this method.
IN	ElementName	string	The value of the ElementName parameter shall be copied into the ElementName property of the CIM_OpaqueManagementData instance to be created by this method.
Required, IN	Owner	CIM_Identity REF	Identity that shall be the owner of the new CIM_OpaqueManagementData instance
IN	BasedOnExtent	CIM_StorageExtent REF	The storage extent on which the new CIM_OpaqueManagementData instance shall bed based
OUT	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the new CIM_OpaqueManagementData instance resulting from the method
OUT	Job	CIM_ConcreteJob REF	Reference to the job (may be null if job completed)

8.2 CIM_OpaqueManagementDataService.Read()

CIM_OpaqueManagementDataService.Read() returns one or more bytes of data from a CIM_OpaqueManagementData instance to the caller.

The return code values for the CIM_OpaqueManagementDataService.Read() method are specified in Table 5.

Table 5 – CIM_OpaqueManagementDataService.Read() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.Read() method are specified in Table 6.

Table 6 – CIM_OpaqueManagementDataService.Read() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found

The parameters for the CIM_OpaqueManagementDataService.Read() method are specified in Table 7.

Table 7 – CIM_OpaqueManagementDataService.Read() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the CIM_OpaqueManagementData instance that represents the data that shall be read
Required, IN	Offset	uint64	The byte offset at which reading shall begin
Required, IN, OUT	Length	uint64	On entry, the number of bytes requested, on exit, the number of bytes returned. If Length is specified as 0 on entry, the method shall fail.
OUT, OctetString	Data	uint8[ ]	The data that were read
IN, OctetString	LockToken	uint8[ ]	A token that represents a Lock placed on the data in the extent referenced by the OpaqueManagementData parameter, using the Lock() method. Parameter should be NULL if Lock() is not supported or if caller does not hold a lock.

8.3 CIM_OpaqueManagementDataService.ExportToURI()

CIM_OpaqueManagementDataService.ExportToURI() copies one or more bytes of data in from a CIM_OpaqueManagementData instance to the Uniform Resource Identifier (URI) specified in the ExportURI parameter.

The return code values for the CIM_OpaqueManagementDataService.ExportToURI() method are specified in Table 8.

Table 8 – CIM_OpaqueManagementDataService.ExportToURI() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred
4096	Job started

Implementation of standard messages is optional. The standard messages for CIM_OpaqueManagementDataService.ExportToURI() are specified in Table 9.

Table 9 – CIM_OpaqueManagementDataService.ExportToURI() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found
WIP219	CIM method not supported by CIM class implementation

The parameters for the CIM_OpaqueManagementDataService.ExportToURI() method are specified in Table 10.

Table 10 – CIM_OpaqueManagementDataService.ExportToURI() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the CIM_OpaqueManagementData instance that represents the data that shall be exported.
Required, IN	Offset	uint64	The byte offset at which reading shall begin
Required, IN, OUT	Length	uint64	On entry, the number of bytes requested, on exit, the number of bytes returned. If Length is specified as 0 on entry, the method shall fail.
Required, IN	ExportURI	string	A URI based on RFC 3986 that indicates the location to which the service shall copy the specified bytes of opaque management data and from which the client may subsequently retrieve them.
IN, OctetString	LockToken	uint8[ ]	A token that represents a Lock placed on the data in the extent referenced by the OpaqueManagementData parameter, using the Lock() method. Parameter should be NULL if Lock() is not supported or if caller does not hold a lock.
OUT	Job	CIM_ConcreteJob REF	Reference to the job (may be null if job completed).

8.4 CIM_OpaqueManagementDataService.Write()

CIM_OpaqueManagementDataService.Write() modifies one or more bytes of data in a CIM_OpaqueManagementData instance.

The return code values for the CIM_OpaqueManagementDataService.Write() method are specified in Table 11.

Table 11 – CIM_OpaqueManagementDataService.Write() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred
4096	Job started

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.Write() method are specified in Table 12.

Table 12 – CIM_OpaqueManagementDataService.Write() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found

The parameters for the CIM_OpaqueManagementDataService.Write() method are specified in Table 13.

Table 13 – CIM_OpaqueManagementDataService.Write() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the CIM_OpaqueManagementData instance that represents the data object that shall be written
Required, IN	Offset	uint64	The byte offset at which writing shall begin. If the value of Offset exceeds the value of the DataSize property in the CIM_OpaqueManagementData instance referenced by the OpaqueManagementData parameter, then this method shall fail.
Required, IN, OUT	Length	uint64	On entry, the number of bytes requested to be written, on exit, the number of bytes actually written. If Length is specified as 0 on entry, the method shall fail.

Qualifiers	Name	Type	Description/Values
IN	Truncate	boolean	If DataSize is greater than Offset + Length, then a value of TRUE shall indicate that the DataSize property value shall be reduced by this operation to Offset + Length, and a value of FALSE shall indicate that the DataSize property value shall remain unchanged by this operation. If DataSize is not greater than Offset + Length, then this parameter shall be ignored.
IN, OctetString	Data	uint8[ ]	The data that to be written to the data object
IN, OctetString	LockToken	uint8[ ]	A token that represents a Lock placed on the data in the extent referenced by the OpaqueManagementData parameter, using the Lock() method. Parameter should be NULL if Lock() is not supported or if caller does not hold a lock.
OUT	Job	CIM_ConcreteJob REF	Reference to the job (may be null if job completed)

8.5 CIM_OpaqueManagementDataService.ImportFromURI()

CIM_OpaqueManagementDataService.ImportFromURI() modifies one or more bytes of data in a CIM_OpaqueManagementData instance. The source data for the operation are retrieved from the URI specified in the ImportURI parameter.

The return code values for CIM_OpaqueManagementDataService.ImportFromURI() are specified in Table 14.

Table 14 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred
4096	Job started

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.ImportFromURI() method are specified in Table 15.

704 **Table 15 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Standard Messages**

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found
WIP219	CIM method not supported by CIM class implementation

705 The parameters for the CIM_OpaqueManagementDataService.ImportFromURI() method are specified in
706 Table 16.

707 **Table 16 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Parameters**

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the CIM_OpaqueManagementData instance that represents the data object that shall be written.
Required, IN	Offset	uint64	The byte offset at which writing shall begin. If the value of Offset exceeds the value of the DataSize property in the CIM_OpaqueManagementData instance referenced by the OpaqueManagementData parameter, then this method shall fail.
Required, IN, OUT	Length	uint64	On entry, the number of bytes requested to be written, on exit, the number of bytes actually written. If Length is specified as 0 on entry, the method shall fail.
IN	Truncate	boolean	If DataSize is greater than Offset + Length, then a value of TRUE shall indicate that the DataSize property value shall be reduced by this operation to Offset + Length, and a value of FALSE shall indicate that the DataSize property value shall remain unchanged by this operation. If DataSize is not greater than Offset + Length, then this parameter shall be ignored.
Required, IN	ImportURI	string	A URI based on RFC 3986 that shall be used by the service to retrieve the bytes of opaque management data to be written into the extent represented by the OpaqueManagementData parameter.
IN, OctetString	LockToken	uint8[ ]	A token that represents a Lock placed on the data in the extent referenced by the OpaqueManagementData parameter, using the Lock() method. Parameter should be NULL if Lock() is not supported or if caller does not hold a lock.
OUT	Job	CIM_ConcreteJob REF	Reference to the job (may be null if job completed)

8.6 CIM_OpaqueManagementDataService.Lock()

CIM_OpaqueManagementDataService.Lock() locks the data in a CIM_OpaqueManagementData instance and returns a LockToken to the caller. Subsequent attempts to access the data in the CIM_OpaqueManagementData instance shall be rejected unless the LockToken is provided. This gives the holder of the LockToken exclusive access to the CIM_OpaqueManagementData instance until CIM_OpaqueManagementDataService.Lock() is called again to unlock it, or until the lock times out, whichever occurs first.

The return code values for the CIM_OpaqueManagementDataService.Lock() method are specified in Table 17.

Table 17 – CIM_OpaqueManagementDataService.Lock() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.ImportFromURI() method are specified in Table 18.

Table 18 – CIM_OpaqueManagementDataService.Lock() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found
WIP219	CIM method not supported by CIM class implementation

The parameters for the CIM_OpaqueManagementDataService.Lock() method are specified in Table 19.

Table 19 – CIM_OpaqueManagementDataService.Lock() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to CIM_OpaqueManagementData instance that shall be locked
Required, IN	Lock	boolean	If true, lock the data from access by another management client, if false, release the data
IN, OUT, OctetString	LockToken	uint8[ ]	A token that shall be returned to the client as output if Lock is True and the return code is 0, or supplied by the client as input if Lock is False. In this way, only the client that set the Lock may remove it or delegate such responsibility to another client.

8.7 CIM_OpaqueManagementDataService.ReassignOwnership()

CIM_OpaqueManagementDataService.ReassignOwnership() transfers ownership of a CIM_OpaqueManagementData instance from one CIM_Identity instance to another.

After successful completion of this method, there shall be exactly one instance of CIM_OpaqueManagementDataOwner that references the CIM_OpaqueManagementData instance referenced by the OpaqueManagementData parameter and the CIM_Identity instance referenced by the NewOwner parameter.

The return code values for the CIM_OpaqueManagementDataService.ReassignOwnership() method are specified in Table 20.

Table 20 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.ReassignOwnership() method are specified in Table 21.

Table 21 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found
WIP219	CIM method not supported by CIM class implementation

The parameters for the CIM_OpaqueManagementDataService.ReassignOwnership() method are specified in Table 22.

Table 22 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	NewOwner	CIM_Identity REF	Reference to the CIM_Identity instance that is the intended new owner of the CIM_OpaqueManagementData instance referenced by the OpaqueManagementData parameter.
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the CIM_OpaqueManagementData instance that the current owner intends to transfer to the CIM_Identity instance referenced by the NewOwner parameter.

8.8 CIM_OpaqueManagementDataService.AssignAccess()

The CIM_OpaqueManagementDataService.AssignAccess() method grants or removes read and/or write access rights to a CIM_OpaqueManagedData instance to a CIM_Identity instance.

If the Activities parameter is not null and not empty, a single instance of CIM_AssociatedPrivilege shall exist between the CIM_Identity instance referenced by the Identity parameter and the CIM_OpaqueManagementData instance referenced by the OpaqueManagementData parameter after successful completion of this method. The Activities array property in this CIM_AssociatedPrivilege instance shall contain 5 (Read) or 6 (Write) if and only if this method's Activities parameter contains 5 (Read) or 6 (Write), respectively. Other values in the Activities array property in this CIM_AssociatedPrivilege instance shall not be affected by this method. If no such a CIM_AssociatedPrivilege instance existed before this method was invoked, the implementation shall create one.

If the Activities parameter is null or empty and a CIM_AssociatedPrivilege instance exists between the instances referenced by the Identity and OpaqueManagementData parameters, then the values 5 (Read) and 6 (Write) shall be removed from the Activities property of this CIM_AssociatedPrivilege instance. If removing these values results in an empty Activities property and the creation class of the CIM_AssociatedPrivilege instance is not CIM_OpaqueManagementDataOwner, then the instance shall be deleted.

The return code values for the CIM_OpaqueManagementDataService.AssignAccess() method are specified in Table 23.

Table 23 – CIM_OpaqueManagementDataService.AssignAccess() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Error occurred

Implementation of standard messages is optional. The standard messages for the CIM_OpaqueManagementDataService.AssignAccess() method are specified in Table 24.

Table 24 – CIM_OpaqueManagementDataService.AssignAccess() Method: Standard Messages

(return) Message ID	Message
WIP212	Invalid CIM method parameter value
WIP213	CIM instance not found
WIP219	CIM method not supported by CIM class implementation

The parameters for the CIM_OpaqueManagementDataService.AssignAccess() method are specified in Table 25.

Table 25 – CIM_OpaqueManagementDataService.AssignAccess() Method: Parameters

Qualifiers	Name	Type	Description/Values
Required, IN	Identity	CIM_Identity REF	Reference to the CIM_Identity instance that is the subject of the privileges modified by this method.
Required, IN	OpaqueManagementData	CIM_OpaqueManagementData REF	Reference to the CIM_OpaqueManagementData instance that is the target of the privileges modified by this method.
Required, IN	Activities[ ]	uint16	The new value of the Activities property of the CIM_AssociatedPrivilege between the CIM_Identity instance referenced by the Identity parameter and the CIM_OpaqueManagementData instance referenced by the OpaqueManagementData parameter. Values are constrained to 5 (Read) and 6 (Write).

8.9 Profile Conventions for Operations

For each profile class (including associations), the implementation requirements for operations, including those in the following default list, are specified in class-specific subclauses of this clause.

The default list of operations is as follows:

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

8.10 CIM_OpaqueManagementDataService Operations

All operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.11 CIM_OpaqueManagementData Operations

Table 26 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 26, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 26 – Operations: CIM_OpaqueManagementData

Operation	Requirement	Messages
ModifyInstance	Mandatory	None

8.11.1 CIM_OpaqueManagementData — ModifyInstance Operation

This clause details the specific requirements for the ModifyInstance operation that is applied to an instance of CIM_OpaqueManagementData.

An implementation shall support reduction of the CIM_OpaqueManagementData.DataSize property through the ModifyInstance operation. An implementation shall accept the ModifyInstance operation if and only if the new value of the DataSize property is less than its current value.

8.12 CIM_OpaqueManagementDataCapabilities Operations

All operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.13 CIM_StorageExtent Operations

All operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.14 CIM_PhysicalMedia Operations

All operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.15 CIM_PhysicalMemory Operations

All operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.16 CIM_HostedService Operations

Table 27 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 27, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 27 – Operations: CIM_HostedService

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

8.17 CIM_ServiceAffectsElement Operations

Table 28 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 28, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 28 – Operations: CIM_ServiceAffectsElement

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

8.18 CIM_ServiceAffectsElementWithQuota Operations

Table 29 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 29, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

The requirements in Table 29 are in addition to those specified for the superclass CIM_ServiceAffectsElement in Table 28.

Table 29 – Operations: CIM_ServiceAffectsElementWithQuota

Operation	Requirement	Messages
CreateInstance	Optional	None
ModifyInstance	Optional	None

8.18.1 CIM_ServiceAffectsElementWithQuota — CreateInstance Operation

This clause details the specific requirements for the CreateInstance operation that is applied to CIM_ServiceAffectsElementWithQuota.

An implementation that supports assignment of quotas shall do so through creation of an instance of CIM_ServiceAffectsElementWithQuota between an instance of CIM_Identity and the central instance.

8.18.2 CIM_ServiceAffectsElementWithQuota — ModifyInstance Operation

This clause details the specific requirements for the ModifyInstance operation that is applied to an instance of CIM_ServiceAffectsElementWithQuota.

An implementation that supports assignment of quotas shall also support modification of quotas through modification of the CIM_ServiceAffectsElementWithQuota.Quota property. Such an implementation shall fail any request to reduce the Quota property to a value less than the value of the QuotaUsed property in the same instance of CIM_ServiceAffectsElementWithQuota.

8.19 CIM_AssociatedPrivilege Operations

Table 30 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 30, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 30 – Operations: CIM_AssociatedPrivilege

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None
DeleteInstance	Conditional	None

8.19.1 CIM_AssociatedPrivilege — DeleteInstance Operation

This clause details the specific requirements for the DeleteInstance operation that is applied to CIM_AssociatedPrivilege.

An implementation that supports assignment of access rights through the CIM_OpaqueManagementDataService.AssignAccess() method shall support deletion of CIM_AssociatedPrivilege instances through the DeleteInstance operation.

8.20 CIM_BasedOn Operations

Table 31 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 31, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 31 – Operations: CIM_BasedOn

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

8.21 CIM_ElementCapabilities Operations

Table 32 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 32, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 32 – Operations: CIM_ElementCapabilities

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

8.22 CIM_RealizesExtent Operations

Table 33 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 33, all operations in the default list in 8.9 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 33 – Operations: CIM_RealizesExtent

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

9 Use Cases

This clause contains object diagrams and use cases for the *Opaque Management Data Profile*. The contents of this clause are for informative purposes only and do not constitute normative requirements for implementations of this specification.

9.1 Base Instances

Figure 2 shows an example of the base instances of an implementation of the *Opaque Management Data Profile*. Support for the *Opaque Management Data Profile* is advertised through the CIM_RegisteredProfile instance rp1, which is associated to the CIM_RegisteredProfile for the *Base Desktop and Mobile Profile* through an instance of CIM_ReferencedProfile.

An instance of the central class CIM_OpaqueManagementDataService is scoped to the CIM_ComputerSystem instance cs0 through an instance of CIM_HostedService.

An instance of CIM_OpaqueManagementDataCapabilities describes the capabilities of the CIM_OpaqueManagementDataService instance to which it is associated through a CIM_ElementCapabilities instance. In this implementation, the total number of bytes of storage (allocated + unallocated) is 16384, with zero bytes allocated to CIM_OpaqueManagementData instances, and the maximum read and write length values are 4096. Note that CIM_OpaqueManagementDataService.AvailableStorage cannot exceed CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage.

The implementation supports the optional class CIM_StorageExtent, which describes the storage extent used for instances of CIM_OpaqueManagementData. The CIM_StorageExtent instance is associated with the CIM_OpaqueManagementDataService instance by an instance of CIM_ServiceAffectsElement.

The DataOrganization property describes the organization of the data in the storage extent. In this instance the organization type is Fixed Block. The storage extent is readable and writeable, and has a BlockSize of 4096 bytes. Note that CIM_StorageExtent.BlockSize contains the same value as CIM_OpaqueManagementDataCapabilities.MaxReadLength and CIM_OpaqueManagementDataCapabilities.MaxWriteLength by coincidence. For each instance of CIM_StorageExtent there are 4 each of the number of blocks available and consumable by the user.

Note that in this system instance there are no instances of CIM_Identity yet associated with the CIM_OpaqueManagementDataService.

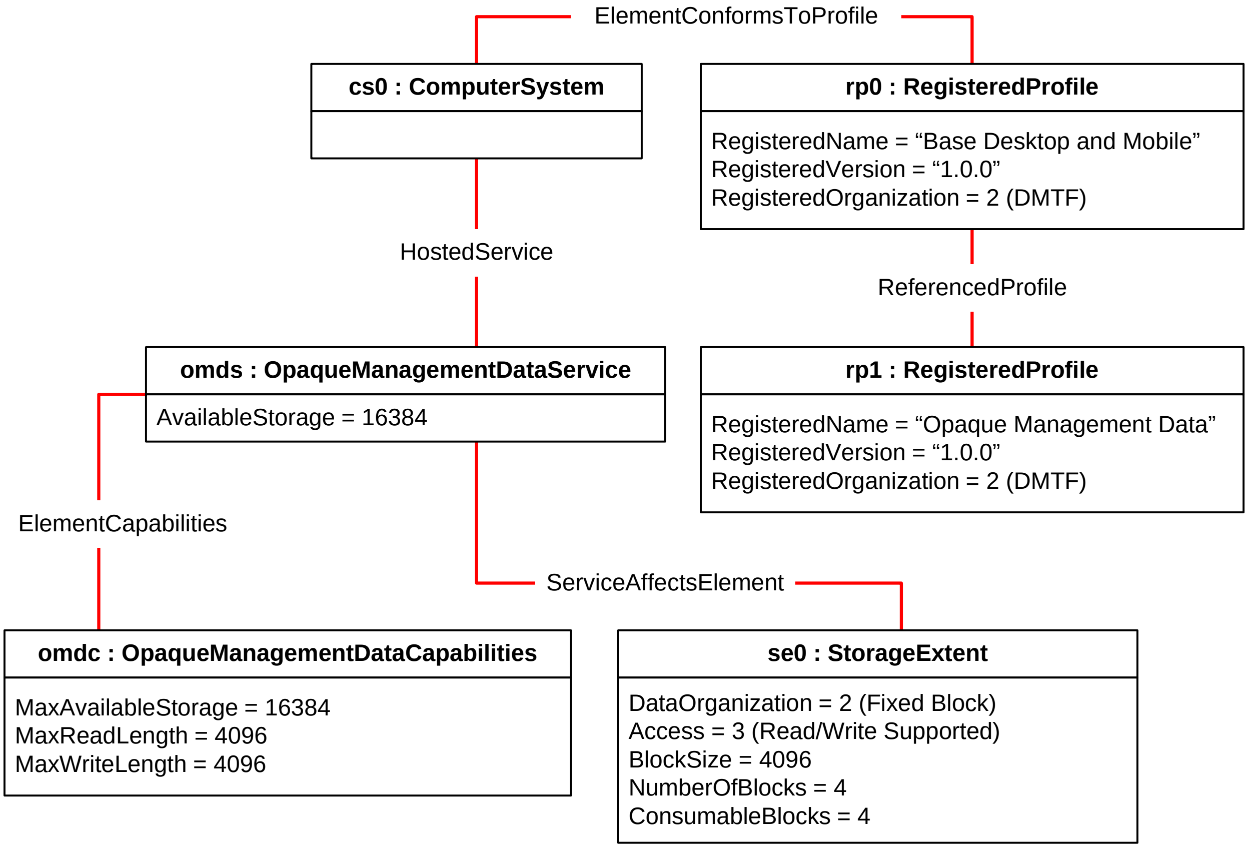
ElementConformsToProfile
cs0 : ComputerSystem
rp0 : RegisteredProfile
RegisteredName = “Base Desktop and Mobile”
RegisteredVersion = “1.0.0”
RegisteredOrganization = 2 (DMTF)
HostedService
ReferencedProfile
omds : OpaqueManagementDataService
AvailableStorage = 16384
rp1 : RegisteredProfile
RegisteredName = “Opaque Management Data”
RegisteredVersion = “1.0.0”
RegisteredOrganization = 2 (DMTF)
ElementCapabilities
ServiceAffectsElement
omdc : OpaqueManagementDataCapabilities
MaxAvailableStorage = 16384
MaxReadLength = 4096
MaxWriteLength = 4096
se0 : StorageExtent
DataOrganization = 2 (Fixed Block)
Access = 3 (Read/Write Supported)
BlockSize = 4096
NumberOfBlocks = 4
ConsumableBlocks = 4

Figure 2 – Base Instances

9.2 CIM_OpaqueManagementData Instances

In Figure 3 the CIM_OpaqueManagementDataService (the service) is managing two instances of CIM_OpaqueManagementData, each of which was created by a client using CIM_OpaqueManagementDataService.Create(). Each instance CIM_OpaqueManagementData has a write limitation, as shown by WriteLimited value = TRUE.

Each user is represented by an instance of CIM_Identity, which is associated to each CIM_OpaqueManagementData instance to which it has access through an instance of CIM_AssociatedPrivilege.

CIM_Identity instance id1 is the owner of CIM_OpaqueManagementData instance omd0 by virtue of having created omd0 using CIM_OpaqueManagementDataService.Create(). Likewise, CIM_Identity instance id0 is the owner of CIM_OpaqueManagementData instance omd1. CIM_Identity instance id0 has granted id1 read and write access to omd1.

Properties in CIM_OpaqueManagementData instances contain values that represent the state and size of the data. DataSize indicates the size of the data, in bytes. The storage used by the OpaqueManagementDataService is represented by the StorageExtent se0, which has a fixed block size of 4096 bytes, and four blocks, all of which are consumable (usable for opaque management data). Each of the OpaqueManagementData instances is connected to this StorageExtent by an instance of the BasedOn association class.

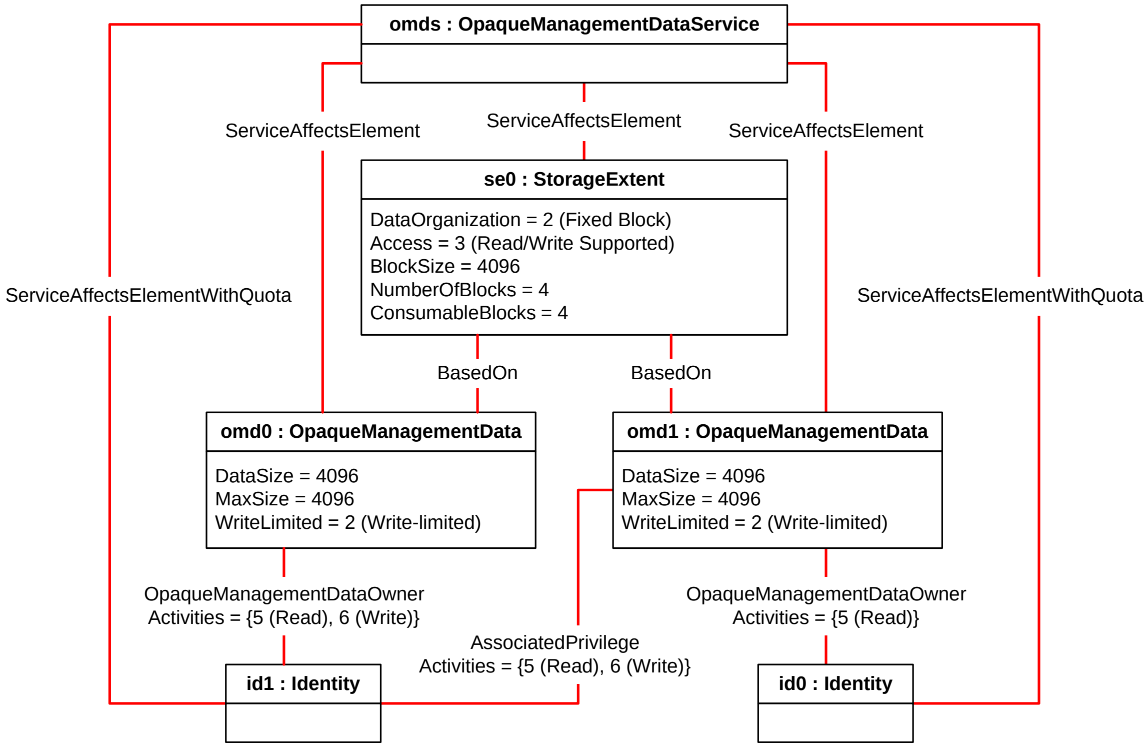
omds : OpaqueManagementDataService
ServiceAffectsElement
ServiceAffectsElement
ServiceAffectsElement
se0 : StorageExtent
DataOrganization = 2 (Fixed Block)
Access = 3 (Read/Write Supported)
BlockSize = 4096
NumberOfBlocks = 4
ConsumableBlocks = 4
ServiceAffectsElementWithQuota
ServiceAffectsElementWithQuota
BasedOn
BasedOn
omd0 : OpaqueManagementData
DataSize = 4096
MaxSize = 4096
WriteLimited = 2 (Write-limited)
omd1 : OpaqueManagementData
DataSize = 4096
MaxSize = 4096
WriteLimited = 2 (Write-limited)
OpaqueManagementDataOwner
Activities = {5 (Read), 6 (Write)}
OpaqueManagementDataOwner
Activities = {5 (Read)}
AssociatedPrivilege
Activities = {5 (Read), 6 (Write)}
id1 : Identity
id0 : Identity

Figure 3 – CIM_OpaqueManagementData Instances

9.3 Physical Modeling

Figure 4 depicts an example of the physical aspects of an *Opaque Management Data Profile* implementation. The central class CIM_OpaqueManagementDataService is instantiated.

The instance of CIM_StorageExtent associated with the CIM_OpaqueManagementDataService instance through CIM_ServiceAffectsElement describes the logical characteristics of the storage used by the service to store opaque management data:

- It has a fixed block size of 64 KB.
- It supports read/write access.
- It consists of 16 blocks, 12 of which are consumable by the service.

The CIM_PhysicalMemory instance associated with the CIM_StorageExtent instance through CIM_Realizes describes the physical aspects of this storage:

- It is a flash device.
- Its capacity is 4 MB. Since this is considerably larger than the associated CIM_StorageExtent, it could contain more than just opaque management data.

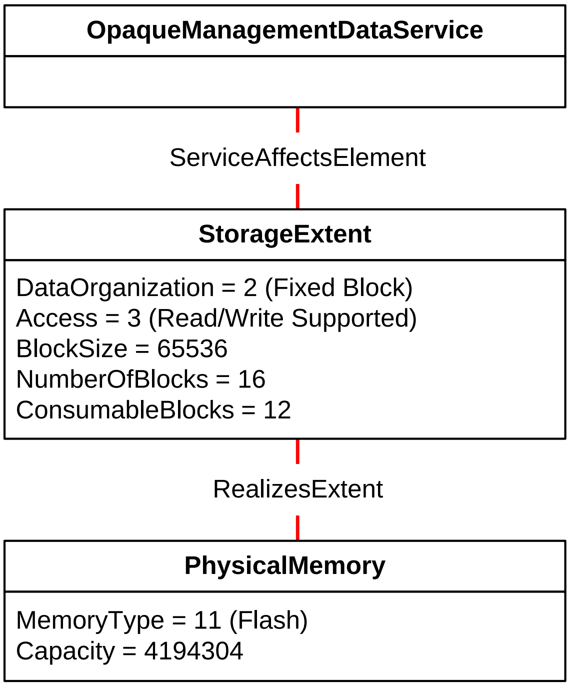

Figure 4 – Physical Modeling

9.4 Create an Instance of CIM_OpaqueManagementData

A client can create an instance of CIM_OpaqueManagementData as follows:

1) Find the CIM_OpaqueManagementDataService and verify it is enabled as follows:

   a) Starting at the CIM_ComputerSystem instance for the managed system, look for instances of CIM_OpaqueManagementDataService associated through CIM_HostedService. If multiple instances of CIM_OpaqueManagementDataService are found, select the one of interest using the property values of each instance.

   b) Verify that CIM_OpaqueManagementDataService.EnabledState property value has the value 2 (Enabled). If the value is 3, then the service is Disabled. If the value is 0 (Unknown), then the service is in transition. For this example, assume EnabledState contains 2 (Enabled).

2) Determine the user identity that will have access to CIM_OpaqueManagementData by finding the instance of CIM_Identity you will use that is associated with CIM_OpaqueManagementDataService through an instance of the CIM_ServiceAffectsElement association.

3) Create an instance of CIM_OpaqueManagementData by invoking CIM_OpaqueManagementService.Create() with required parameters. If CIM_OpaqueManagementDataService.Create() completes successfully, the new CIM_OpaqueManagementData instance will be associated with the owning CIM_Identity instance through a new instance of CIM_OpaqueManagementDataOwner.

9.5 Read Data Represented by an Instance of CIM_OpaqueManagementData

A client can use CIM_OpaqueManagementDataService.Read() or CIM_OpaqueManagementDataService.ExportToURI() to read from an instance of CIM_OpaqueManagementData.

Each method contains IN parameters that indicate the starting read byte offset and the number of bytes to return, which is limited by CIM_OpaqueManagementDataCapabilities.MaxReadLength. If the client needs to read more data, it can use the method CIM_OpaqueManagementDataService.Lock() to block access to the data by other clients while it completes multiple read operations.

Note that CIM_OpaqueManagementDataService.ExportToURI() will read from the storage represented by the CIM_OpaqueManagementData instance and write to the URI supplied.

9.6 Write Data Represented by an Instance of CIM_OpaqueManagementData

A client can use CIM_OpaqueManagementDataService.Write() or CIM_OpaqueManagementDataService.ImportFromURI() to write to an instance of CIM_OpaqueManagementData.

Each method contains IN parameters that indicate the starting write byte offset and the number of bytes to write, which is limited by CIM_OpaqueManagementDataCapabilities.MaxWriteLength. If the client needs to write more data, it can use the method CIM_OpaqueManagementDataService.Lock() to block access to the data by other clients while it completes multiple write operations.

Note that CIM_OpaqueManagementDataService.ImportFromURI() will read from the URI supplied and write to the storage represented by the CIM_OpaqueManagementData instance.

9.7 Overwrite and Truncate Data Represented by an Instance of CIM_OpaqueManagementData

A client can use the CIM_OpaqueManagementDataService to write to an instance of CIM_OpaqueManagementData and truncate previously written data using the methods described in clause 9.6. Each of these methods includes a Truncate parameter. When Truncate = TRUE, previously written data that falls after the last byte of new data shall be erased from storage.

9.8 Delete an Instance of CIM_OpaqueManagementData

The deletion of a single instance of a class is realized by using the intrinsic operation DeleteInstance() as defined in DSP0200 version 1.2.

9.9 Enumerate Instances of CIM_OpaqueManagementData

Enumeration of a class object is supported via EnumerateInstance() as defined in DSP0200 version 1.2.

9.10 Manage CIM_OpaqueManagementData Access

If the CIM_OpaqueManagementDataCapabilities.SupportedMethods property contains the value 9 (AssignAccess) then the CIM_OpaqueManagementData.AssignAccess() method can be used to manage access to CIM_OpaqueManagementData instances.

10 CIM Elements

Table 34 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 34. Clauses 7 (“Implementation “) and 8 (“Methods”) may impose additional requirements on these elements.

Table 34 – CIM Elements: Opaque Management Data Profile

Element Name	Requirement	Description
Classes		
CIM_OpaqueManagementDataService	Mandatory	See 7.1 and 10.1.
CIM_OpaqueManagementData	Mandatory	See 7.3 and 10.2.
CIM_OpaqueManagementDataCapabilities	Mandatory	See 7.2 and 10.3.
CIM_StorageExtent	Optional	See 7.3.10 and 10.4.
CIM_PhysicalMedia	Optional	See 7.3.11 and 10.5.
CIM_PhysicalMemory	Optional	See 7.3.11 and 10.6.
CIM_RegisteredProfile	Mandatory	See 10.7.
CIM_HostedService	Mandatory	See 7.1.6 and 10.8.
CIM_ServiceAffectsElement	Optional	See 7.3.13.1, 10.9, 10.10, and 10.11.
CIM_ServiceAffectsElementWithQuota	Optional	See 7.3.13.2.1 and 10.12.
CIM_AssociatedPrivilege	Optional	See 7.3.4.1 and 10.13.
CIM_OpaqueManagementDataOwner	Mandatory	See 7.3.3.1 and 10.14.
CIM_BasedOn	Optional	See 7.3.10 and 10.15.
CIM_ElementCapabilities	Mandatory	See 7.2 and 10.16.
CIM_RealizesExtent	Optional	See 7.3.11 and 10.17.
Indications		
None defined in this profile		

10.1 CIM_OpaqueManagementDataService

Table 35 contains requirements for elements of CIM_OpaqueManagementDataService. These requirements are in addition to those specified for the superclass CIM_EnabledLogicalElement in the *Enabled Logical Element Profile*.

Table 35 – Class: CIM_OpaqueManagementDataService

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
LockTimeout	Conditional	See 7.1.1.
MaxCreateSize	Optional	See 7.1.2.

Elements	Requirement	Notes
AvailableStorage	Mandatory	See 7.1.3.
Create()	Optional	See 8.1.
Read()	Mandatory	See 8.2.
ExportToURI()	Optional	See 8.3.
Write()	Mandatory	See 8.4.
ImportFromURI()	Optional	See 8.5.
Lock()	Optional	See 8.6.
ReassignOwnership()	Optional	See 8.7.
AssignAccess()	Optional	See 8.8.

10.2 CIM_OpaqueManagementData

Table 36 contains requirements for elements of CIM_OpaqueManagementData. The requirements in Table 36 are in addition to those specified for the superclass CIM_StorageExtent in Table 38.

Table 36 – Class: CIM_OpaqueManagementData

Elements	Requirement	Notes
DataSize	Mandatory	See 7.3.5.
MaxSize	Mandatory	See 7.3.6.
DataFormat	Mandatory	See 7.3.7.
LastAccessed	Optional	See 7.3.8.
WriteLimited	Conditional	See 7.3.9.

10.3 CIM_OpaqueManagementDataCapabilities

Table 37 contains requirements for elements of CIM_OpaqueManagementDataCapabilities. These requirements are in addition to those specified for the superclass CIM_EnabledLogicalElementCapabilities in the *Enabled Logical Element Profile*.

Table 37 – Class: CIM_OpaqueManagementDataCapabilities

Elements	Requirement	Notes
MaxAvailableStorage	Mandatory	See 7.2.1.
MaxReadLength	Mandatory	See 7.2.2.
MaxWriteLength	Mandatory	See 7.2.3.
SupportedMethods	Mandatory	See 7.2.4.
SupportedExportURISchemes	Conditional	See 7.2.5.
SupportedImportURISchemes	Conditional	See 7.2.6.

10.4 CIM_StorageExtent

Table 38 contains requirements for elements of CIM_StorageExtent. These requirements are in addition to those specified for superclass CIM_EnabledLogicalElement in the *Enabled Logical Element Profile*.

Table 38 – Class: CIM_StorageExtent

Elements	Requirement	Notes
DataOrganization	Optional	None
Access	Optional	None
BlockSize	Optional	None
NumberOfBlocks	Optional	None
ConsumableBlocks	Optional	None
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
DeviceID	Mandatory	**Key**

10.5 CIM_PhysicalMedia

Table 39 contains the requirements for elements of CIM_PhysicalMedia.

Table 39 – Class: CIM_PhysicalMedia

Elements	Requirement	Notes
Tag	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
Capacity	Mandatory	None
MediaType	Mandatory	None
Manufacturer	Conditional	See the *Physical Asset Profile*.
Model	Conditional	See the *Physical Asset Profile*.
Serial Number	Conditional	See the *Physical Asset Profile*.
PartNumber	Conditional	See the *Physical Asset Profile*.
SKU	Conditional	See the *Physical Asset Profile*.
CanBeFRUed	Optional	Should be implemented when the PhysicalMedia can be field replaced.
ElementName	Mandatory	Matches pattern ".*".

10.6 CIM_PhysicalMemory

An implementation that supports CIM_PhysicalMemory as specified in clause 7.3.11 shall support the CIM_PhysicalMemory element requirements defined in the *Physical Asset Profile*.

10.7 CIM_RegisteredProfile

An implementation shall support CIM_RegisteredProfile. With the exception of the mandatory values specified for the properties in Table 40, the behavior of the CIM_RegisteredProfile instance is defined by the *Profile Registration Profile*.

Table 40 – Class: CIM_RegisteredProfile

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of “Opaque Management Data”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

10.8 CIM_HostedService

Table 41 contains the requirements for elements of CIM_HostedService.

Table 41 – Class: CIM_HostedService

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementDataService. Cardinality 1..*

10.9 CIM_ServiceAffectsElement (CIM_OpaqueManagementData)

Table 42 contains the element requirements for CIM_ServiceAffectsElement between CIM_OpaqueManagementDataService and CIM_OpaqueManagementData.

Table 42 – Class: CIM_ServiceAffectsElement

Elements	Requirement	Notes
AffectedElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementData. Cardinality *
AffectingElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementDataService. Cardinality 1
ElementEffects	Mandatory	Contains 5 (Manages)

10.10 CIM_ServiceAffectsElement (CIM_StorageExtent)

Table 43 contains the element requirements for CIM_ServiceAffectsElement between CIM_OpaqueManagementDataService and CIM_StorageExtent.

Table 43 – Class: CIM_ServiceAffectsElement

Elements	Requirement	Notes
AffectedElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_StorageExtent. Cardinality *
AffectingElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementDataService. Cardinality 1
ElementEffects	Mandatory	Contains 5 (Manages)

10.11 CIM_ServiceAffectsElement (CIM_Identity)

Table 44 contains the element requirements for CIM_ServiceAffectsElement between CIM_OpaqueManagementDataService and CIM_Identity.

Table 44 – Class: CIM_ServiceAffectsElement

Elements	Requirement	Notes
AffectedElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_Identity. Cardinality *
AffectingElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementDataService. Cardinality 1
ElementEffects	Mandatory	Contains 5 (Manages)

10.12 CIM_ServiceAffectsElementWithQuota

Table 45 contains the requirements for elements of CIM_ServiceAffectsElementWithQuota. The requirements in Table 45 are in addition to those specified for the superclass CIM_ServiceAffectsElement in Table 42.

Table 45 – Class: CIM_ServiceAffectsElementWithQuota

Elements	Requirement	Notes
Quota	Conditional	See 7.3.13.2.1.1.
QuotaUsed	Conditional	See 7.3.13.2.1.2.
QuotaUnits	Conditional	See 7.3.13.2.1.3.

10.13 CIM_AssociatedPrivilege

Table 46 contains the requirements for elements of CIM_AssociatedPrivilege.

Table 46 – Class: CIM_AssociatedPrivilege

Elements	Requirement	Notes
Target	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementData. Cardinality *
Subject	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_Identity. Cardinality *
Activities	Mandatory	See clause 7.3.4.1.1.

10.14 CIM_OpaqueManagementDataOwner

No properties are required beyond those required for CIM_AssociatedPrivilege in 10.13.

10.15 CIM_BasedOn

Table 47 contains the requirements for elements of CIM_BasedOn.

Table 47 – Class: CIM_BasedOn

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_StorageExtent or one of its subclasses. Cardinality 0..1
Dependent	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementData. Cardinality *

10.16 CIM_ElementCapabilities

Table 48 contains the requirements for elements of CIM_ElementCapabilities.

Table 48 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
ManagedElement	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementDataService. Cardinality 1..*
Capabilities	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_OpaqueManagementDataCapabilities. Cardinality 0..1

10.17 CIM_RealizesExtent

Table 49 contains the requirements for elements of CIM_RealizesExtent.

Table 49 – Class: CIM_RealizesExtent

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_PhysicalMedia or CIM_PhysicalMemory. Cardinality 0..1
Dependent	Mandatory	**Key:** This property shall contain a reference to an instance of CIM_StorageExtent. Cardinality *
StartingAddress	Optional	None.

ANNEX A
(Informative)

Change Log

Version	Date	Description
1.0.0	05-20-2009	DMTF Standard Release
1.0.1	10-21-2009	Corrections to Figure 1