



1  
2  
3  
4

**Document Identifier: DSP1070**

**Date: 2014-05-22**

**Version: 1.0.2**

## 5 **Opaque Management Data Profile**

6 **Document Type: Specification**  
7 **Document Status: DMTF Standard**  
8 **Document Language: en-US**  
9

10 Copyright Notice

11 Copyright © 2008, 2014 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

12 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
 13 management and interoperability. Members and non-members may reproduce DMTF specifications and  
 14 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to  
 15 time, the particular version and release date should always be noted.

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 26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

32

# CONTENTS

34	Foreword .....	7
35	Introduction.....	8
36	1 Scope .....	9
37	2 Normative References.....	9
38	2.1 Approved References .....	9
39	2.2 Other References.....	9
40	3 Terms and Definitions .....	9
41	4 Symbols and Abbreviated Terms .....	11
42	5 Synopsis .....	11
43	6 Description .....	11
44	7 Implementation Requirements .....	13
45	7.1 CIM_OpaqueManagementDataService .....	13
46	7.2 CIM_OpaqueManagementDataCapabilities .....	14
47	7.3 Representing Opaque Management Data .....	16
48	8 Methods.....	21
49	8.1 CIM_OpaqueManagementDataService.Create() .....	21
50	8.2 CIM_OpaqueManagementDataService.Read().....	22
51	8.3 CIM_OpaqueManagementDataService.ExportToURI() .....	23
52	8.4 CIM_OpaqueManagementDataService.Write().....	25
53	8.5 CIM_OpaqueManagementDataService.ImportFromURI() .....	26
54	8.6 CIM_OpaqueManagementDataService.Lock().....	27
55	8.7 CIM_OpaqueManagementDataService.ReassignOwnership() .....	28
56	8.8 CIM_OpaqueManagementDataService.AssignAccess() .....	29
57	8.9 Profile Conventions for Operations.....	31
58	8.10 CIM_OpaqueManagementDataService Operations .....	31
59	8.11 CIM_OpaqueManagementData Operations .....	31
60	8.12 CIM_OpaqueManagementDataCapabilities Operations .....	31
61	8.13 CIM_StorageExtent Operations .....	31
62	8.14 CIM_PhysicalMedia Operations .....	32
63	8.15 CIM_PhysicalMemory Operations .....	32
64	8.16 CIM_HostedService Operations .....	32
65	8.17 CIM_ServiceAffectsElement Operations .....	32
66	8.18 CIM_ServiceAffectsElementWithQuota Operations .....	32
67	8.19 CIM_AssociatedPrivilege Operations .....	33
68	8.20 CIM_BasedOn Operations.....	34
69	8.21 CIM_ElementCapabilities Operations.....	34
70	8.22 CIM_RealizesExtent Operations.....	34
71	9 Use Cases .....	35
72	9.1 Base Instances .....	35
73	9.2 CIM_OpaqueManagementData Instances .....	36
74	9.3 Physical Modeling .....	37
75	9.4 Create an Instance of CIM_OpaqueManagementData .....	38
76	9.5 Read Data Represented by an Instance of CIM_OpaqueManagementData .....	38
77	9.6 Write Data Represented by an Instance of CIM_OpaqueManagementData .....	39
78	9.7 Overwrite and Truncate Data Represented by an Instance of	
79	CIM_OpaqueManagementData .....	39
80	9.8 Delete an Instance of CIM_OpaqueManagementData.....	39
81	9.9 Enumerate Instances of CIM_OpaqueManagementData.....	39
82	9.10 Manage CIM_OpaqueManagementData Access .....	39
83	10 CIM Elements .....	40
84	10.1 CIM_OpaqueManagementDataService.....	40

85 10.2 CIM\_OpaqueManagementData ..... 41

86 10.3 CIM\_OpaqueManagementDataCapabilities ..... 41

87 10.4 CIM\_StorageExtent ..... 42

88 10.5 CIM\_PhysicalMedia ..... 42

89 10.6 CIM\_PhysicalMemory ..... 42

90 10.7 CIM\_RegisteredProfile ..... 43

91 10.8 CIM\_HostedService ..... 43

92 10.9 CIM\_ServiceAffectsElement (CIM\_OpaqueManagementData) ..... 43

93 10.10 CIM\_ServiceAffectsElement (CIM\_StorageExtent) ..... 44

94 10.11 CIM\_ServiceAffectsElement (CIM\_Identity) ..... 44

95 10.12 CIM\_ServiceAffectsElementWithQuota ..... 44

96 10.13 CIM\_AssociatedPrivilege ..... 45

97 10.14 CIM\_OpaqueManagementDataOwner ..... 45

98 10.15 CIM\_BasedOn ..... 45

99 10.16 CIM\_ElementCapabilities ..... 45

100 10.17 CIM\_RealizesExtent ..... 46

101 ANNEX A (Informative) Change Log ..... 47

102

103 **Figures**

104 Figure 1 – Opaque Management Data Profile: Profile Class Diagram ..... 12

105 Figure 2 – Base Instances ..... 36

106 Figure 3 – CIM\_OpaqueManagementData Instances ..... 37

107 Figure 4 – Physical Modeling ..... 38

108

109 **Tables**

110 Table 1 – Related Profiles ..... 11

111 Table 2 – CIM\_OpaqueManagementDataService.Create() Method: Return Code Values ..... 21

112 Table 3 – CIM\_OpaqueManagementDataService.Create() Method: Standard Messages ..... 21

113 Table 4 – CIM\_OpaqueManagementDataService.Create() Method: Parameters ..... 22

114 Table 5 – CIM\_OpaqueManagementDataService.Read() Method: Return Code Values ..... 23

115 Table 6 – CIM\_OpaqueManagementDataService.Read() Method: Standard Messages ..... 23

116 Table 7 – CIM\_OpaqueManagementDataService.Read() Method: Parameters ..... 23

117 Table 8 – CIM\_OpaqueManagementDataService.ExportToURI() Method: Return Code Values ..... 24

118 Table 9 – CIM\_OpaqueManagementDataService.ExportToURI() Method: Standard Messages ..... 24

119 Table 10 – CIM\_OpaqueManagementDataService.ExportToURI() Method: Parameters ..... 24

120 Table 11 – CIM\_OpaqueManagementDataService.Write() Method: Return Code Values ..... 25

121 Table 12 – CIM\_OpaqueManagementDataService.Write() Method: Standard Messages ..... 25

122 Table 13 – CIM\_OpaqueManagementDataService.Write() Method: Parameters ..... 25

123 Table 14 – CIM\_OpaqueManagementDataService.ImportFromURI() Method: Return Code Values ..... 26

124 Table 15 – CIM\_OpaqueManagementDataService.ImportFromURI() Method: Standard Messages ..... 26

125 Table 16 – CIM\_OpaqueManagementDataService.ImportFromURI() Method: Parameters ..... 27

126 Table 17 – CIM\_OpaqueManagementDataService.Lock() Method: Return Code Values ..... 28

127 Table 18 – CIM\_OpaqueManagementDataService.Lock() Method: Standard Messages ..... 28

128 Table 19 – CIM\_OpaqueManagementDataService.Lock() Method: Parameters ..... 28

129 Table 20 – CIM\_OpaqueManagementDataService.ReassignOwnership() Method: Return Code Values ..... 29

130 Table 21 – CIM\_OpaqueManagementDataService.ReassignOwnership() Method: Standard Messages ..... 29

131 Table 22 – CIM\_OpaqueManagementDataService.ReassignOwnership() Method: Parameters ..... 29

132 Table 23 – CIM\_OpaqueManagementDataService.AssignAccess() Method: Return Code Values ..... 30

133 Table 24 – CIM\_OpaqueManagementDataService.AssignAccess() Method: Standard Messages..... 30

134 Table 25 – CIM\_OpaqueManagementDataService.AssignAccess() Method: Parameters ..... 30

135 Table 26 – Operations: CIM\_OpaqueManagementData ..... 31

136 Table 27 – Operations: CIM\_HostedService ..... 32

137 Table 28 – Operations: CIM\_ServiceAffectsElement ..... 32

138 Table 29 – Operations: CIM\_ServiceAffectsElementWithQuota ..... 33

139 Table 30 – Operations: CIM\_AssociatedPrivilege ..... 33

140 Table 31 – Operations: CIM\_BasedOn ..... 34

141 Table 32 – Operations: CIM\_ElementCapabilities ..... 34

142 Table 33 – Operations: CIM\_RealizesExtent..... 34

143 Table 34 – CIM Elements: Opaque Management Data Profile..... 40

144 Table 35 – Class: CIM\_OpaqueManagementDataService ..... 40

145 Table 36 – Class: CIM\_OpaqueManagementData ..... 41

146 Table 37 – Class: CIM\_OpaqueManagementDataCapabilities ..... 41

147 Table 38 – Class: CIM\_StorageExtent..... 42

148 Table 39 – Class: CIM\_PhysicalMedia ..... 42

149 Table 40 – Class: CIM\_RegisteredProfile ..... 43

150 Table 41 – Class: CIM\_HostedService ..... 43

151 Table 42 – Class: CIM\_ServiceAffectsElement ..... 43

152 Table 43 – Class: CIM\_ServiceAffectsElement ..... 44

153 Table 44 – Class: CIM\_ServiceAffectsElement ..... 44

154 Table 45 – Class: CIM\_ServiceAffectsElementWithQuota ..... 44

155 Table 46 – Class: CIM\_AssociatedPrivilege ..... 45

156 Table 47 – Class: CIM\_BasedOn ..... 45

157 Table 48 – Class: CIM\_ElementCapabilities..... 45

158 Table 49 – Class: CIM\_RealizesExtent ..... 46

159



161

## Foreword

162 The *Opaque Management Data Profile* (DSP1070) was prepared by the Server Desktop Mobile Platforms  
163 Working Group of the DMTF.

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

## 166 Acknowledgments

167 The authors wish to acknowledge the following people.

168 Editor:

- 169 • David Hines – Intel
- 170 • Hemal Shah - Broadcom

171 Contributors:

- 172 • Murali Rajagopal
- 173 •
- 174 • Sharon L Smith – Intel

175 Participants from the DMTF Desktop and Mobile Working Group:

- 176 • Bob Blair – AMD
- 177 • Jon Hass – Dell
- 178 • Jeff Hilland – HP
- 179 • Joel Clark – Intel
- 180 • Steve Hand – Symantec

181

182

## Introduction

183 The information in this specification should be sufficient for a provider or consumer of this data to identify  
184 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to  
185 represent and manage opaque management data that is modeled using the DMTF Common Information  
186 Model (CIM) core and extended model definitions.

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



189

# Opaque Management Data Profile

## 190 1 Scope

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

## 193 2 Normative References

194 The following referenced documents are indispensable for the application of this document. For dated  
195 references, only the edition cited applies. For undated references, the latest edition of the referenced  
196 document (including any amendments) applies.

### 197 2.1 Approved References

198 DMTF DSP0004, *CIM Infrastructure Specification 2.5*,  
199 [http://www.dmtf.org/standards/published\\_documents/DSP0004\\_2.5.pdf](http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf)

200 DMTF DSP0200, *CIM Operations over HTTP 1.2*,  
201 <http://www.dmtf.org/sites/default/files/standards/documents/DSP200.html>

202 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,  
203 [http://www.dmtf.org/standards/published\\_documents/DSP1001\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf)

204 DMTF DSP1011, *Physical Asset Profile 1.0*,  
205 [http://www.dmtf.org/standards/published\\_documents/DSP1011\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1011_1.0.pdf)

206 DMTF DSP1033, *Profile Registration Profile 1.0*,  
207 [http://www.dmtf.org/standards/published\\_documents/DSP1033\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf)

208 DMTF DSP1034, *Simple Identity Management Profile 1.0*,  
209 [http://www.dmtf.org/standards/published\\_documents/DSP1034\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1034_1.0.pdf)

210 DMTF DSP1058, *Base Desktop and Mobile Profile 1.0*,  
211 [http://www.dmtf.org/standards/published\\_documents/DSP1058\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1058_1.0.pdf)

212 DMTF DSP1080, *Enabled Logical Element Profile 1.0*,  
213 [http://www.dmtf.org/standards/published\\_documents/DSP1080\\_1.0.pdf](http://www.dmtf.org/standards/published_documents/DSP1080_1.0.pdf)

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

### 216 2.2 Other References

217 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,  
218 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>

## 219 3 Terms and Definitions

220 For the purposes of this document, the following terms and definitions apply. For the purposes of this  
221 document, the terms and definitions given in [DSP1033](#) and [DSP1001](#) also apply.

- 222 **3.1**
- 223 **can**
- 224 used for statements of possibility and capability, whether material, physical, or causal
- 225 **3.2**
- 226 **cannot**
- 227 used for statements of possibility and capability, whether material, physical, or causal
- 228 **3.3**
- 229 **conditional**
- 230 indicates requirements to be followed strictly in order to conform to the document when the specified
- 231 conditions are met
- 232 **3.4**
- 233 **mandatory**
- 234 indicates requirements to be followed strictly in order to conform to the document and from which no
- 235 deviation is permitted
- 236 **3.5**
- 237 **may**
- 238 indicates a course of action permissible within the limits of the document
- 239 **3.6**
- 240 **need not**
- 241 indicates a course of action permissible within the limits of the document
- 242 **3.7**
- 243 **optional**
- 244 indicates a course of action permissible within the limits of the document
- 245 **3.8**
- 246 **referencing profile**
- 247 indicates a profile that owns the definition of this class and can include a reference to this profile in its
- 248 "Related Profiles" table
- 249 **3.9**
- 250 **shall**
- 251 indicates requirements to be followed strictly in order to conform to the document and from which no
- 252 deviation is permitted
- 253 **3.10**
- 254 **shall not**
- 255 indicates requirements to be followed in order to conform to the document and from which no deviation is
- 256 permitted
- 257 **3.11**
- 258 **should**
- 259 indicates that among several possibilities, one is recommended as particularly suitable, without
- 260 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 261 **3.12**
- 262 **should not**
- 263 indicates that a certain possibility or course of action is deprecated but not prohibited

264 **3.13**  
 265 **unspecified**  
 266 indicates that this profile does not define any constraints for the referenced CIM element or operation

267 **4 Symbols and Abbreviated Terms**

268 The following symbols and abbreviations are used in this document.

269 **4.1**  
 270 **CIM**  
 271 Common Information Model

272 **5 Synopsis**

273 **Profile Name:** Opaque Management Data

274 **Version:** 1.0.2

275 **Organization:** DMTF

276 **CIM Schema Version:** 2.22

277 **Central Class:** CIM\_OpaqueManagementDataService

278 **Scoping Class:** CIM\_ComputerSystem

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

281 CIM\_OpaqueManagementDataService shall be the Central Class of this profile. The instance(s) of  
 282 CIM\_OpaqueManagementDataService shall be the Central Instance(s) of this profile.  
 283 CIM\_ComputerSystem shall be the Scoping Class of this profile. The instance of CIM\_ComputerSystem  
 284 with which the Central Instance is associated through an instance of CIM\_HostedService shall be the  
 285 Scoping Instance of this profile.

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

287 **Table 1 – Related Profiles**

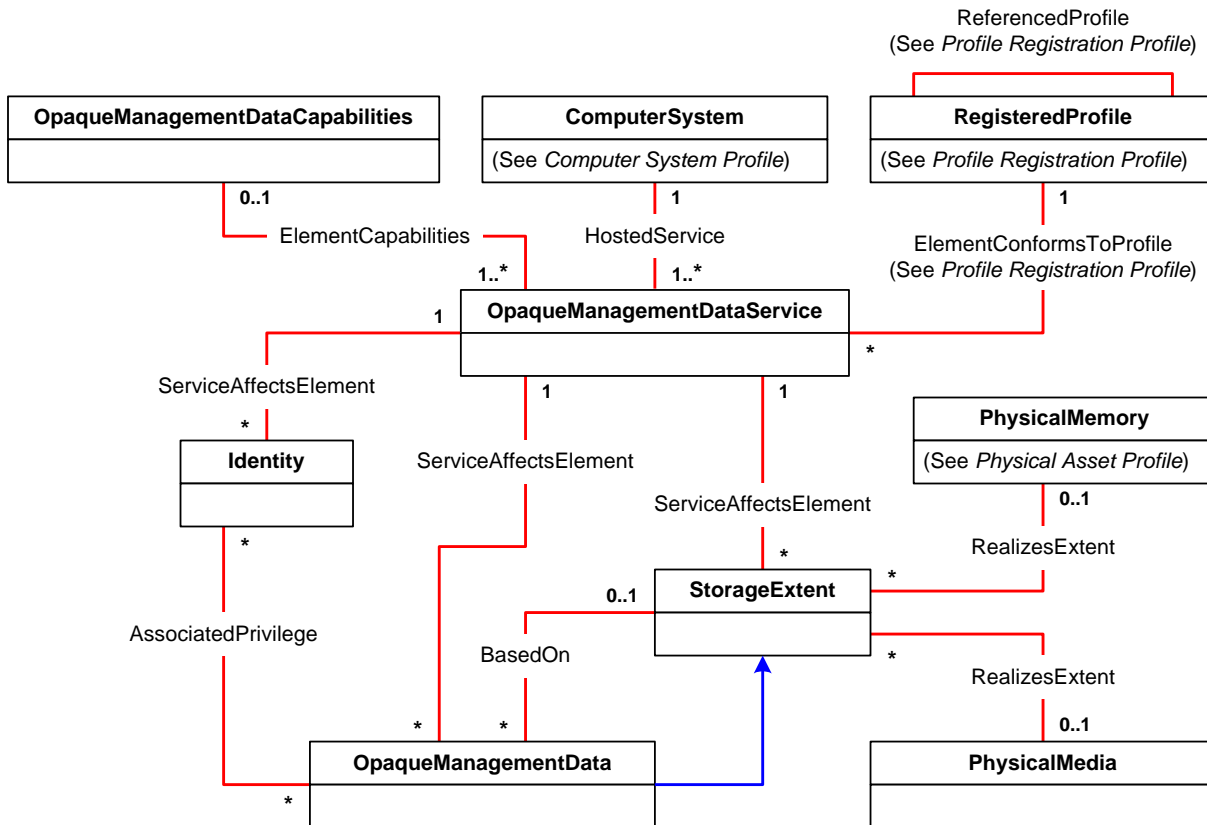
Profile Name	Organization	Version	Requirement
Profile Registration	DMTF	1.0	Mandatory
Simple Identity Management	DMTF	1.0	Mandatory
Physical Asset	DMTF	1.0	Optional
Enabled Logical Element	DMTF	1.0	Mandatory

288 **6 Description**

289 The *Opaque Management Data Profile* defines capabilities for producing, consuming, and describing  
 290 “opaque” management data in managed systems. The structure and semantics of opaque management  
 291 data are known to management client software, but may not be known to the implementation. Opaque  
 292 data are stored in non-volatile memory that is accessible to the implementation. Management clients are  
 293 able to read and write opaque data, as well as manage its lifetime and access rights. The non-volatile  
 294 storage is often used as a “mailbox”, with local management components producing data for consumption

295 by remote management clients, or vice versa. The information in this specification should be sufficient for  
 296 a provider or consumer of this data to identify unambiguously the classes, properties, methods, and  
 297 values that must be instantiated and manipulated to represent and manage the opaque management  
 298 data of systems and subsystems modeled using the DMTF CIM core and extended model definitions.

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



301

302 **Figure 1 – Opaque Management Data Profile: Profile Class Diagram**

303 The central class is CIM\_OpaqueManagementDataService, which supports extrinsic methods for  
 304 creating, reading, writing, and locking opaque management data. CIM\_OpaqueManagementDataService  
 305 is associated to the scoping class through CIM\_HostedService.

306 CIM\_OpaqueManagementDataService capabilities are described by  
 307 CIM\_OpaqueManagementDataCapabilities, a specialization of CIM\_EnabledLogicalElementCapabilities.  
 308 CIM\_OpaqueManagementDataCapabilities is associated with CIM\_OpaqueManagementDataService  
 309 through CIM\_ElementCapabilities.

310 Logical units of opaque management data are represented by CIM\_OpaqueManagementData, which  
 311 contains properties that describe, but do not access the opaque management data. All access to opaque  
 312 management data itself is done through CIM\_OpaqueManagementDataService extrinsic methods.  
 313 CIM\_OpaqueManagementData is a specialization of CIM\_StorageExtent.

314 CIM\_OpaqueManagementData is managed by CIM\_OpaqueManagementDataService. This relationship  
 315 is represented through the association class CIM\_ServiceAffectsElement.

316 CIM\_OpaqueManagementData may be based on a larger CIM\_StorageExtent. This relationship is  
317 represented through the association class CIM\_BasedOn. The impact of storing opaque management  
318 data on this larger extent is in turn represented through the association class CIM\_ServiceAffectsElement  
319 between CIM\_StorageExtent and CIM\_OpaqueManagementDataService.

320 Owners and users of CIM\_OpaqueManagementData are represented by CIM\_Identity. The relationship  
321 between CIM\_OpaqueManagementData and its users and owners is represented by  
322 CIM\_AssociatedPrivilege and its subclass CIM\_OpaqueManagementDataOwner, respectively.

323 The CIM\_OpaqueManagementDataService may impose limits on the amount of storage users may  
324 allocate for opaque management data by assigning a quota, represented by  
325 CIM\_ServiceAffectsElementWithQuota.

326 The physical aspects of the storage used for opaque management data are represented by  
327 CIM\_PhysicalMemory or CIM\_PhysicalMedia, depending on the type of physical storage used. These are  
328 associated to CIM\_StorageExtent through CIM\_RealizesExtent.

329 Support for the *Opaque Management Data Profile* is advertised by CIM\_RegisteredProfile.

## 330 7 Implementation Requirements

331 This clause specifies the requirements related to the arrangement of instances and properties for  
332 implementations of this profile. Methods are specified in clause 8, and properties are specified in clause  
333 10.

### 334 7.1 CIM\_OpaqueManagementDataService

335 At least one instance of CIM\_OpaqueManagementDataService shall be supported.

#### 336 7.1.1 CIM\_OpaqueManagementDataService.LockTimeout

337 The CIM\_OpaqueManagementDataService extrinsic method Lock() allows a client to lock an instance of  
338 CIM\_OpaqueManagementData.

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

343 If an implementation supports locking CIM\_OpaqueManagementData and supports an automatic lock  
344 timeout, then CIM\_OpaqueManagementDataService.LockTimeout shall be supported and shall contain  
345 the duration of this timeout.

346 If an implementation does not support locking CIM\_OpaqueManagementData or does not support an  
347 automatic lock timeout, then CIM\_OpaqueManagementDataService.LockTimeout shall be NULL.

#### 348 7.1.2 CIM\_OpaqueManagementDataService.MaxCreateSize

349 CIM\_OpaqueManagementDataService.MaxCreateSize may be supported. If supported,  
350 CIM\_OpaqueManagementDataService.MaxCreateSize shall contain the largest value that the  
351 implementation will currently accept in the MaxSize parameter in a call to  
352 CIM\_OpaqueManagementDataService.Create().

### 353 7.1.3 CIM\_OpaqueManagementDataService.AvailableStorage

354 CIM\_OpaqueManagementDataService.AvailableStorage shall contain the total number of free bytes of  
355 storage currently available for allocation to CIM\_OpaqueManagementData instances. Accordingly, the  
356 number of bytes of storage already allocated to CIM\_OpaqueManagementData instances shall be equal  
357 to the difference between CIM\_OpaqueManagementDataCapabilities.MaxAvailableStorage and  
358 CIM\_OpaqueManagementDataService.AvailableStorage.

359 CIM\_OpaqueManagementDataService.AvailableStorage shall not exceed  
360 CIM\_OpaqueManagementDataCapabilities.MaxAvailableStorage.

### 361 7.1.4 Managing CIM\_OpaqueManagementDataService.ElementName

362 An implementation may support management of the ElementName property of  
363 CIM\_OpaqueManagementDataService. The abstract [Enabled Logical Element Profile](#) specifies  
364 requirements for supporting ElementName management in subclasses of CIM\_EnabledLogicalElement.  
365 The implementation of CIM\_OpaqueManagementDataService shall meet the requirements in the [Enabled](#)  
366 [Logical Element Profile](#), with CIM\_OpaqueManagementDataService in place of  
367 CIM\_EnabledLogicalElement and CIM\_OpaqueManagementDataCapabilities in place of  
368 CIM\_EnabledLogicalElementCapabilities.

### 369 7.1.5 Managing CIM\_OpaqueManagementDataService State

370 An implementation may support management of CIM\_OpaqueManagementDataService state. The  
371 abstract [Enabled Logical Element Profile](#) specifies requirements for supporting state management in  
372 subclasses of CIM\_EnabledLogicalElement. The implementation of  
373 CIM\_OpaqueManagementDataService shall meet the requirements of the [Enabled Logical Element](#)  
374 [Profile](#), with CIM\_OpaqueManagementDataService in place of CIM\_EnabledLogicalElement and  
375 CIM\_OpaqueManagementDataCapabilities in place of CIM\_EnabledLogicalElementCapabilities.

### 376 7.1.6 CIM\_HostedService

377 Each instance of CIM\_OpaqueManagementDataService shall be associated with the scoping instance of  
378 CIM\_ComputerSystem through an instance of CIM\_HostedService.

## 379 7.2 CIM\_OpaqueManagementDataCapabilities

380 Exactly one instance of CIM\_OpaqueManagementDataCapabilities shall be associated with each  
381 instance of CIM\_OpaqueManagementDataService through the CIM\_ElementCapabilities association.

### 382 7.2.1 CIM\_OpaqueManagementDataCapabilities.MaxAvailableStorage

383 CIM\_OpaqueManagementDataCapabilities.MaxAvailableStorage shall contain the total number of bytes  
384 of storage for opaque management data managed by the instance of  
385 CIM\_OpaqueManagementDataService associated with the CIM\_OpaqueManagementDataCapabilities  
386 instance through CIM\_ElementCapabilities. This number shall include all allocated and unallocated  
387 storage.

### 388 7.2.2 CIM\_OpaqueManagementDataCapabilities.MaxReadLength

389 CIM\_OpaqueManagementDataCapabilities.MaxReadLength shall contain the maximum value that the  
390 implementation supports in the Length parameter of the CIM\_OpaqueManagementDataService.Read()  
391 and CIM\_OpaqueManagementDataService.ExportToURI() methods. The value of  
392 CIM\_OpaqueManagementDataCapabilities.MaxReadLength may be less than the value in the DataSize  
393 property of a CIM\_OpaqueManagementData instance. A client that wishes to read more bytes than the  
394 value of CIM\_OpaqueManagementDataCapabilities.MaxReadLength should do so by using  
395 CIM\_OpaqueManagementDataService.Read() multiple times.

### 396 7.2.3 CIM\_OpaqueManagementDataCapabilities.MaxWriteLength

397 CIM\_OpaqueManagementDataCapabilities.MaxWriteLength shall contain the maximum value that the  
398 implementation supports in the Length parameter of the CIM\_OpaqueManagementDataService.Write()  
399 and and CIM\_OpaqueManagementDataService.ImportFromURI() methods. The value of  
400 CIM\_OpaqueManagementDataCapabilities.MaxWriteLength may be less than the value in the MaxSize  
401 property of a CIM\_OpaqueManagementData instance. A client that wishes to write more bytes than the  
402 value of CIM\_OpaqueManagementDataCapabilities.MaxWriteLength should do so by using  
403 CIM\_OpaqueManagementDataService.Write() multiple times.

### 404 7.2.4 CIM\_OpaqueManagementDataCapabilities.SupportedMethods

405 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain each value that corresponds  
406 to a CIM\_OpaqueManagementDataService method supported by the implementation, and shall not  
407 contain any value that corresponds to a CIM\_OpaqueManagementDataService method not supported by  
408 the implementation.

409 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 2 (Create) if and  
410 only if the implementation supports the method CIM\_OpaqueManagementDataService.Create().

411 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 3 (Read) if and  
412 only if the implementation supports the method CIM\_OpaqueManagementDataService.Read().

413 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 4 (ExportToURI) if  
414 and only if the implementation supports the method  
415 CIM\_OpaqueManagementDataService.ExportToURI().

416 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 5 (Write) if and  
417 only if the implementation supports the method CIM\_OpaqueManagementDataService.Write().

418 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 6 (ImportFromURI)  
419 if and only if the implementation supports the method  
420 CIM\_OpaqueManagementDataService.ImportFromURI().

421 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 7 (Lock) if and only  
422 if the implementation supports the method CIM\_OpaqueManagementDataService.Lock().

423 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 8  
424 (ReassignOwnership) if and only if the implementation supports the method  
425 CIM\_OpaqueManagementDataService.ReassignOwnership().

426 CIM\_OpaqueManagementDataCapabilities.SupportedMethods shall contain the value 9 (AssignAccess) if  
427 and only if the implementation supports the method  
428 CIM\_OpaqueManagementDataService.AssignAccess().

### 429 7.2.5 CIM\_OpaqueManagementDataCapabilities.SupportedExportURISchemes

430 The CIM\_OpaqueManagementDataCapabilities.SupportedExportURISchemes property shall list the URI  
431 schemes that are supported by the implementation for.  
432 CIM\_OpaqueManagementDataService.ExportToURI().

### 433 7.2.6 CIM\_OpaqueManagementDataCapabilities.SupportedImportURISchemes

434 The CIM\_OpaqueManagementDataCapabilities.SupportedImportURISchemes property shall list the URI  
435 schemes that are supported by the implementation for.  
436 CIM\_OpaqueManagementDataService.ImportFromURI().

## 437 7.3 Representing Opaque Management Data

438 Zero or more instances of CIM\_OpaqueManagementData shall be instantiated. Each  
439 CIM\_OpaqueManagementData instance shall represent an independent logical unit of opaque  
440 management data.

### 441 7.3.1 Managing CIM\_OpaqueManagementData.ElementName

442 An implementation may support management of the ElementName property of  
443 CIM\_OpaqueManagementData. The abstract [Enabled Logical Element Profile](#) specifies requirements for  
444 supporting ElementName management in subclasses of CIM\_EnabledLogicalElement. The  
445 implementation of CIM\_OpaqueManagementData shall meet the requirements in the [Enabled Logical  
446 Element Profile](#), with CIM\_OpaqueManagementData in place of CIM\_EnabledLogicalElement.

### 447 7.3.2 Managing CIM\_OpaqueManagementData State

448 An implementation may support management of CIM\_OpaqueManagementData state. The abstract  
449 [Enabled Logical Element Profile](#) specifies requirements for supporting state management in subclasses  
450 of CIM\_EnabledLogicalElement. The implementation of CIM\_OpaqueManagementData shall meet the  
451 requirements of the [Enabled Logical Element Profile](#), with CIM\_OpaqueManagementData in place of  
452 CIM\_EnabledLogicalElement.

## 453 7.3.3 Ownership of Opaque Management Data

454 Each instance of CIM\_OpaqueManagementData shall be owned by exactly one user, represented by an  
455 instance of CIM\_Identity. The owner of each CIM\_OpaqueManagementData instance created by a client  
456 using CIM\_OpaqueManagementDataService.Create() shall be the CIM\_Identity instance referenced by  
457 the Owner parameter in that method.

### 458 7.3.3.1 CIM\_OpaqueManagementDataOwner

459 Every instance of CIM\_OpaqueManagementData shall have exactly one owner. Exactly one instance of  
460 CIM\_OpaqueManagementDataOwner shall exist between each instance of  
461 CIM\_OpaqueManagementData and the CIM\_Identity instance that owns it. The requirements defined in  
462 clause 7.3.4.1 for instances of CIM\_AssociatedPrivilege shall also be met for instances of its subclass  
463 CIM\_OpaqueManagementDataOwner.

### 464 7.3.3.2 Reassignment of Ownership

465 An implementation may support reassignment of ownership of a CIM\_OpaqueManagementData instance  
466 by its owner CIM\_Identity instance to another CIM\_Identity instance.

#### 467 7.3.3.2.1 Reassignment of Ownership Is Supported

468 If an implementation supports reassignment of ownership of a CIM\_OpaqueManagementData instance  
469 by its owner CIM\_Identity instance to another CIM\_Identity instance, it shall meet the requirements  
470 specified in this clause.

##### 471 7.3.3.2.1.1 CIM\_OpaqueManagementDataService.ReassignOwnership()

472 The CIM\_OpaqueManagementDataService.ReassignOwnership() method shall be implemented and shall  
473 not return 1 (Not supported).

##### 474 7.3.3.2.1.2 CIM\_OpaqueManagementDataCapabilities.SupportedMethods

475 The CIM\_OpaqueManagementDataCapabilities.SupportedMethods array property shall contain the value  
476 8 (ReassignOwnership).



#### 477 **7.3.3.2.2 Reassignment of Ownership Is Not Supported**

478 If an implementation does not support reassignment of ownership of a CIM\_OpaqueManagementData  
479 instance by its owner CIM\_Identity instance to another CIM\_Identity instance, it shall meet the  
480 requirements specified in this clause.

##### 481 **7.3.3.2.2.1 CIM\_OpaqueManagementDataService.ReassignOwnership()**

482 The CIM\_OpaqueManagementDataService.ReassignOwnership() method shall not be implemented.

##### 483 **7.3.3.2.2.2 CIM\_OpaqueManagementDataCapabilities.SupportedMethods**

484 The CIM\_OpaqueManagementDataCapabilities.SupportedMethods array property shall not contain the  
485 value 8 (ReassignOwnership).

### 486 **7.3.4 Access Rights to Opaque Management Data**

487 An implementation shall support read and write access to opaque management data, where “reading”  
488 corresponds to executing CIM\_OpaqueManagementDataService.Read() or  
489 CIM\_OpaqueManagementDataService.ExportToURI() and “writing” corresponds to executing  
490 CIM\_OpaqueManagementDataService.Write() or  
491 CIM\_OpaqueManagementDataService.ImportOpaqueManagementDataToURI().

#### 492 **7.3.4.1 CIM\_AssociatedPrivilege**

493 An instance of CIM\_AssociatedPrivilege or its subclass CIM\_OpaqueManagementDataOwner shall exist  
494 between each CIM\_OpaqueManagementData instance and each CIM\_Identity instance that has access  
495 to it. If the CIM\_Identity instance is the owner of the CIM\_OpaqueManagementData instance, this shall be  
496 an instance of CIM\_OpaqueManagementDataOwner. If the CIM\_Identity instance is not the owner of the  
497 CIM\_OpaqueManagementData instance, this shall be an instance of CIM\_AssociatedPrivilege.

##### 498 **7.3.4.1.1 CIM\_AssociatedPrivilege.Activities**

499 Read access and write access shall be represented by the values 5 (Read), and 6 (Write), respectively, in  
500 CIM\_AssociatedPrivilege.Activities.

501 The Activities property shall contain the value 5 (Read) if and only if the CIM\_Identity instance is  
502 permitted to execute CIM\_OpaqueManagementDataService.Read() and  
503 CIM\_OpaqueManagementDataService.ExportToURI() on the associated CIM\_OpaqueManagementData.  
504 The Activities property shall not contain the value 5 (Read) in more than one element.

505 The Activities property shall contain the value 6 (Write) if and only if the CIM\_Identity instance is permitted  
506 to execute CIM\_OpaqueManagementDataService.Write() and  
507 CIM\_OpaqueManagementDataService.ImportFromURI() on the associated  
508 CIM\_OpaqueManagementData instance. The Activities property shall not contain the value 6 (Write) in  
509 more than one element.

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

##### 511 **7.3.4.1.2 CIM\_AssociatedPrivilege.ActivityQualifiers**

512 The value “DMTF: DSP1070:” shall appear in each entry that corresponds to an entry in  
513 CIM\_AssociatedPrivilege.Activities that conforms to clause 7.3.4.1.1.

##### 514 **7.3.4.1.3 CIM\_AssociatedPrivilege.QualifierFormats**

515 The value 12 (Specification-defined) shall appear in each entry that corresponds to an entry in  
516 CIM\_AssociatedPrivilege.Activities that conforms to clause 7.3.4.1.1.

### 517 7.3.4.2 Assignment of Access Rights

518 An implementation may support assignment and removal of read and write access rights on a  
519 CIM\_OpaqueManagementData instance by its owner CIM\_Identity instance to any CIM\_Identity  
520 instances, including itself.

#### 521 7.3.4.2.1 Assignment of Access Rights Is Supported

522 If an implementation supports assignment of read and write access rights on a  
523 CIM\_OpaqueManagementData instance by its owner CIM\_Identity instance to other CIM\_Identity  
524 instances, it shall meet the requirements specified in this clause.

##### 525 7.3.4.2.1.1 CIM\_OpaqueManagementDataService.AssignAccess()

526 The CIM\_OpaqueManagementDataService.AssignAccess() method shall be implemented and shall not  
527 return 1 (Not supported).

##### 528 7.3.4.2.1.2 CIM\_OpaqueManagementDataCapabilities.SupportedMethods

529 The CIM\_OpaqueManagementDataCapabilities.SupportedMethods array property shall contain the value  
530 9 (AssignAccess).

#### 531 7.3.4.2.2 Assignment of Access Rights Is Not Supported

532 If an implementation does not support assignment of read and write access rights on a  
533 CIM\_OpaqueManagementData instance by its owner CIM\_Identity instance to other CIM\_Identity  
534 instances, it shall meet the requirements specified in this clause.

##### 535 7.3.4.2.2.1 CIM\_OpaqueManagementDataService.AssignAccess()

536 The CIM\_OpaqueManagementDataService.AssignAccess() method shall not be implemented.

##### 537 7.3.4.2.2.2 CIM\_OpaqueManagementDataCapabilities.SupportedMethods

538 The CIM\_OpaqueManagementDataCapabilities.SupportedMethods array property shall not contain the  
539 value 9 (AssignAccess).

### 540 7.3.5 CIM\_OpaqueManagementData.DataSize

541 CIM\_OpaqueManagementData.DataSize shall contain the number of logically contiguous bytes of  
542 meaningful data in the CIM\_OpaqueManagementData instance, starting at offset 0, and ending at offset  
543 DataSize – 1.

544 The initial value of DataSize upon creation of the CIM\_OpaqueManagementData instance and before any  
545 writes shall be 0.

546 The value of DataSize shall not be greater than the value in CIM\_OpaqueManagementData.MaxSize.

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

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

### 551 7.3.6 CIM\_OpaqueManagementData.MaxSize

552 CIM\_OpaqueManagementData.MaxSize shall contain the maximum number of total bytes that can be  
553 written into an instance of CIM\_OpaqueManagementData.

### 554 **7.3.7 CIM\_OpaqueManagementData.DataFormat**

555 CIM\_OpaqueManagementData.DataFormat shall contain a string that describes the format of the data in  
556 an instance of CIM\_OpaqueManagementData.

### 557 **7.3.8 CIM\_OpaqueManagementData.LastAccessed**

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

### 561 **7.3.9 CIM\_OpaqueManagementData.WriteLimited**

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

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

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

### 571 **7.3.10 Representing the Storage of Opaque Management Data**

572 An implementation may support the use of CIM\_StorageExtent to describe the storage in which  
573 CIM\_OpaqueManagementData instances are stored. If an implementation does support  
574 CIM\_StorageExtent for this purpose, each CIM\_OpaqueManagementData instance shall be associated  
575 with an instance of CIM\_StorageExtent through an instance of CIM\_BasedOn.

### 576 **7.3.11 Representing the Physical Realization of Opaque Management Data**

577 An implementation may support the use of CIM\_PhysicalMedia or CIM\_PhysicalMemory to describe the  
578 physical realization of CIM\_OpaqueManagementData instances or the CIM\_StorageExtent upon which  
579 there are based. If an implementation does support CIM\_PhysicalMedia or CIM\_PhysicalMemory for this  
580 purpose, each CIM\_StorageExtent instance shall be associated with an instance of CIM\_PhysicalMedia  
581 or CIM\_PhysicalMemory through an instance of CIM\_Realizes.

### 582 **7.3.12 Representing Management of Opaque Management Data**

583 Each instance of CIM\_OpaqueManagementData shall be associated with exactly one instance of  
584 CIM\_OpaqueManagementDataService through CIM\_ServiceAffectsElement or its subclass  
585 CIM\_ServiceAffectsElementWithQuota.

586 Each instance of CIM\_StorageExtent that is associated with one or more instances of  
587 CIM\_OpaqueManagementData through CIM\_BasedOn shall be associated through an instance of  
588 CIM\_ServiceAffectsElement with the instance of CIM\_OpaqueManagementDataService that is  
589 associated with the CIM\_OpaqueManagementData instance through CIM\_ServiceAffectsElement or its  
590 subclass CIM\_ServiceAffectsElementWithQuota.

### 591 **7.3.13 Representing Management of Users of Opaque Management Data**

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

**594 7.3.13.1 CIM\_ServiceAffectsElement**

595 Each instance of CIM\_Identity that represents a user of opaque management data shall be connected  
596 through an instance of CIM\_ServiceAffectsElement to each instance of  
597 CIM\_OpaqueManagementDataService that is associated to at least one instance of  
598 CIM\_OpaqueManagementData that is used by the user.

**599 7.3.13.2 Representing Opaque Management Data Storage Quotas**

600 An implementation may support assignment of opaque management data storage quotas to CIM\_Identity  
601 instances. The CIM\_ServiceAffectsElement specialization CIM\_ServiceAffectsElementWithQuota shall be  
602 used to represent such quotas. If an implementation supports assignment of opaque management data  
603 storage quotas to CIM\_Identity instances, the requirements in this clause shall be met.

**604 7.3.13.2.1 CIM\_ServiceAffectsElementWithQuota**

605 The creation class of the CIM\_ServiceAffectsElement instance between the CIM\_Identity instance and  
606 the CIM\_OpaqueManagementDataService instance that assigns the quota shall be  
607 CIM\_ServiceAffectsElementWithQuota.

**608 7.3.13.2.1.1 CIM\_ServiceAffectsElementWithQuota.Quota**

609 CIM\_ServiceAffectsElementWithQuota.Quota shall contain the quota value, in bytes, that has been  
610 assigned to the CIM\_Identity instance associated through CIM\_ServiceAffectsElementWithQuota to the  
611 instance of CIM\_OpaqueManagementDataService.

**612 7.3.13.2.1.2 CIM\_ServiceAffectsElementWithQuota.QuotaUsed**

613 CIM\_ServiceAffectsElementWithQuota.QuotaUsed shall contain the number of bytes of the quota value  
614 that have been used by the CIM\_Identity instance associated through  
615 CIM\_ServiceAffectsElementWithQuota to the instance of CIM\_OpaqueManagementDataService.

**616 7.3.13.2.1.3 CIM\_ServiceAffectsElementWithQuota.QuotaUnits**

617 CIM\_ServiceAffectsElementWithQuota.QuotaUnits shall contain “byte”.

**618 7.3.14 Locking Opaque Management Data**

619 An implementation may support concurrent access to CIM\_OpaqueManagementData instances by  
620 multiple clients. If the CIM\_OpaqueManagementDataService.Read() and  
621 CIM\_OpaqueManagementDataService.Write() operations of multiple clients on a single  
622 CIM\_OpaqueManagementData instance are interlaced, data incoherence or corruption may result.

623 To avoid data incoherence and corruption, an implementation may support a lock on each  
624 CIM\_OpaqueManagementData instance.

**625 7.3.14.1 Locking Opaque Management Data Is Supported**

626 The requirements in this clause shall be met by an implementation that supports locking  
627 CIM\_OpaqueManagementData instances.

628 The CIM\_OpaqueManagementDataService method Lock() shall be implemented and shall not return 1  
629 (Not supported).

**630 7.3.14.2 Locking Opaque Management Data Is Not Supported**

631 The requirements in this clause shall be met by an implementation that does not support locking  
632 CIM\_OpaqueManagementData instances.

633 The CIM\_OpaqueManagementDataService method Lock() shall not be implemented, or shall be  
 634 implemented and always return 1 (Not supported).

635 **8 Methods**

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

638 **8.1 CIM\_OpaqueManagementDataService.Create()**

639 If an implementation supports creation of CIM\_OpaqueManagementData instances by clients, it shall  
 640 support the CIM\_OpaqueManagementDataService.Create() method. The client that invokes this method  
 641 shall be the initial owner of the CIM\_OpaqueManagementData instance that it creates.

642 If the CIM\_OpaqueManagementDataService.Create() method succeeds, it shall create the following  
 643 instances:

- 644 1) A single CIM\_OpaqueManagementData instance.
- 645 2) A single CIM\_OpaqueManagementDataOwner instance between the new  
 646 CIM\_OpaqueManagementData instance and the CIM\_Identity instance referenced by the  
 647 Owner parameter of the method. The Activities properties of the new  
 648 CIM\_OpaqueManagementDataOwner instance shall contain the values 5 (Read) and 6 (Write).

649 The return code values for the CIM\_OpaqueManagementDataService.Create() method are specified in  
 650 Table 2.

651 **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

652 Implementation of standard messages is optional. The standard messages for the  
 653 CIM\_OpaqueManagementDataService.Create() method are specified in Table 3.

654 **Table 3 – CIM\_OpaqueManagementDataService.Create() Method: Standard Messages**

(return) Message ID	Message Name
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found
WIPG0219	Method not supported by class implementation

655 The parameters for the CIM\_OpaqueManagementDataService.Create() method are specified in Table 4.

656 **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 be 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)

657 **8.2 CIM\_OpaqueManagementDataService.Read()**

658 CIM\_OpaqueManagementDataService.Read() returns one or more bytes of data from a  
659 CIM\_OpaqueManagementData instance to the caller.

660 The return code values for the CIM\_OpaqueManagementDataService.Read() method are specified in  
661 Table 5.

662 **Table 5 – CIM\_OpaqueManagementDataService.Read() Method: Return Code Values**

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

663 Implementation of standard messages is optional. The standard messages for the  
 664 CIM\_OpaqueManagementDataService.Read() method are specified in Table 6.

665 **Table 6 – CIM\_OpaqueManagementDataService.Read() Method: Standard Messages**

(return) Message ID	Message Name
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found

666 The parameters for the CIM\_OpaqueManagementDataService.Read() method are specified in Table 7.

667 **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.

668 **8.3 CIM\_OpaqueManagementDataService.ExportToURI()**

669 CIM\_OpaqueManagementDataService.ExportToURI() copies one or more bytes of data in from a  
 670 CIM\_OpaqueManagementData instance to the Uniform Resource Identifier (URI) specified in the  
 671 ExportURI parameter.

672 The return code values for the CIM\_OpaqueManagementDataService.ExportToURI() method are  
 673 specified in Table 8.

674 **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

675 Implementation of standard messages is optional. The standard messages for  
 676 CIM\_OpaqueManagementDataService.ExportToURI() are specified in Table 9.

677 **Table 9 – CIM\_OpaqueManagementDataService.ExportToURI() Method: Standard Messages**

(return) Message ID	Message Name
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found
WIPG0219	Method not supported by class implementation

678 The parameters for the CIM\_OpaqueManagementDataService.ExportToURI() method are specified in  
 679 Table 10.

680 **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).



681 **8.4 CIM\_OpaqueManagementDataService.Write()**

682 CIM\_OpaqueManagementDataService.Write() modifies one or more bytes of data in a  
 683 CIM\_OpaqueManagementData instance.

684 The return code values for the CIM\_OpaqueManagementDataService.Write() method are specified in  
 685 Table 11.

686 **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

687 Implementation of standard messages is optional. The standard messages for the  
 688 CIM\_OpaqueManagementDataService.Write() method are specified in Table 12.

689 **Table 12 – CIM\_OpaqueManagementDataService.Write() Method: Standard Messages**

(return) Message ID	Message
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found

690 The parameters for the CIM\_OpaqueManagementDataService.Write() method are specified in Table 13.

691 **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. If the sum of Offset and Length exceeds OpaqueManagementData.MaxSize, then 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)

692 **8.5 CIM\_OpaqueManagementDataService.ImportFromURI()**

693 CIM\_OpaqueManagementDataService.ImportFromURI() modifies one or more bytes of data in a  
694 CIM\_OpaqueManagementData instance. The source data for the operation are retrieved from the URI  
695 specified in the ImportURI parameter.

696 The return code values for CIM\_OpaqueManagementDataService.ImportFromURI() are specified in  
697 Table 14.

698 **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

699 Implementation of standard messages is optional. The standard messages for the  
700 CIM\_OpaqueManagementDataService.ImportFromURI() method are specified in Table 15.

701 **Table 15 – CIM\_OpaqueManagementDataService.ImportFromURI() Method: Standard Messages**

(return) Message ID	Message
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found
WIPG0219	Method not supported by class implementation

702 The parameters for the CIM\_OpaqueManagementDataService.ImportFromURI() method are specified in  
 703 Table 16.

704 **Table 16 – CIM\_OpaqueManagementDataService.ImportFromURI() Method: Parameters**

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_OpaqueManagement Data 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)

705 **8.6 CIM\_OpaqueManagementDataService.Lock()**

706 CIM\_OpaqueManagementDataService.Lock() locks the data in a CIM\_OpaqueManagementData  
 707 instance and returns a LockToken to the caller. Subsequent attempts to access the data in the  
 708 CIM\_OpaqueManagementData instance shall be rejected unless the LockToken is provided. This gives  
 709 the holder of the LockToken exclusive access to the CIM\_OpaqueManagementData instance until  
 710 CIM\_OpaqueManagementDataService.Lock() is called again to unlock it, or until the lock times out,  
 711 whichever occurs first.

712 The return code values for the CIM\_OpaqueManagementDataService.Lock() method are specified in  
713 Table 17.

714 **Table 17 – CIM\_OpaqueManagementDataService.Lock() Method: Return Code Values**

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

715 Implementation of standard messages is optional. The standard messages for the  
716 CIM\_OpaqueManagementDataService.ImportFromURI() method are specified in Table 18.

717 **Table 18 – CIM\_OpaqueManagementDataService.Lock() Method: Standard Messages**

(return) Message ID	Message
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found
WIPG0219	Method not supported by CIM class implementation

718 The parameters for the CIM\_OpaqueManagementDataService.Lock() method are specified in Table 19.

719 **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.

720 **8.7 CIM\_OpaqueManagementDataService.ReassignOwnership()**

721 CIM\_OpaqueManagementDataService.ReassignOwnership() transfers ownership of a  
722 CIM\_OpaqueManagementData instance from one CIM\_Identity instance to another.

723 After successful completion of this method, there shall be exactly one instance of  
724 CIM\_OpaqueManagementDataOwner that references the CIM\_OpaqueManagementData instance  
725 referenced by the OpaqueManagementData parameter and the CIM\_Identity instance referenced by the  
726 NewOwner parameter.

727 The return code values for the CIM\_OpaqueManagementDataService.ReassignOwnership() method are  
728 specified in Table 20.

729 **Table 20 – CIM\_OpaqueManagementDataService.ReassignOwnership() Method: Return Code Values**

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

730 Implementation of standard messages is optional. The standard messages for the  
 731 CIM\_OpaqueManagementDataService.ReassignOwnership() method are specified in Table 21.

732 **Table 21 – CIM\_OpaqueManagementDataService.ReassignOwnership() Method: Standard Messages**

(return) Message ID	Message
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found
WIPG0219	Method not supported by class implementation

733 The parameters for the CIM\_OpaqueManagementDataService.ReassignOwnership() method are  
 734 specified in Table 22.

735 **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.

736 **8.8 CIM\_OpaqueManagementDataService.AssignAccess()**

737 The CIM\_OpaqueManagementDataService.AssignAccess() method grants or removes read and/or write  
 738 access rights to a CIM\_OpaqueManagedData instance to a CIM\_Identity instance.

739 If the Activities parameter is not null and not empty, a single instance of CIM\_AssociatedPrivilege shall  
 740 exist between the CIM\_Identity instance referenced by the Identity parameter and the  
 741 CIM\_OpaqueManagementData instance referenced by the OpaqueManagementData parameter after  
 742 successful completion of this method. The Activities array property in this CIM\_AssociatedPrivilege  
 743 instance shall contain 5 (Read) or 6 (Write) if and only if this method's Activities parameter contains 5  
 744 (Read) or 6 (Write), respectively. Other values in the Activities array property in this  
 745 CIM\_AssociatedPrivilege instance shall not be affected by this method. If no such a  
 746 CIM\_AssociatedPrivilege instance existed before this method was invoked, the implementation shall  
 747 create one.

748 If the Activities parameter is null or empty and a CIM\_AssociatedPrivilege instance exists between the  
 749 instances referenced by the Identity and OpaqueManagementData parameters, then the values 5 (Read)

750 and 6 (Write) shall be removed from the Activities property of this CIM\_AssociatedPrivilege instance. If  
 751 removing these values results in an empty Activities property and the creation class of the  
 752 CIM\_AssociatedPrivilege instance is not CIM\_OpaqueManagementDataOwner, then the instance shall  
 753 be deleted.

754 The return code values for the CIM\_OpaqueManagementDataService.AssignAccess() method are  
 755 specified in Table 23.

756 **Table 23 – CIM\_OpaqueManagementDataService.AssignAccess() Method: Return Code Values**

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

757 Implementation of standard messages is optional. The standard messages for the  
 758 CIM\_OpaqueManagementDataService.AssignAccess() method are specified in Table 24.

759 **Table 24 – CIM\_OpaqueManagementDataService.AssignAccess() Method: Standard Messages**

(return) Message ID	Message
WIPG0249	Invalid input parameter value
WIPG0213	Instance not found
WIPG0219	Method not supported by class implementation

760 The parameters for the CIM\_OpaqueManagementDataService.AssignAccess() method are specified in  
 761 Table 25.

762 **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).

763 **8.9 Profile Conventions for Operations**

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

766 The default list of operations is as follows:

- 767 • GetInstance
- 768 • Associators
- 769 • AssociatorNames
- 770 • References
- 771 • ReferenceNames
- 772 • EnumerateInstances
- 773 • EnumerateInstanceNames

774 **8.10 CIM\_OpaqueManagementDataService Operations**

775 All operations in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

777 **8.11 CIM\_OpaqueManagementData Operations**

778 Table 26 lists implementation requirements for operations. If implemented, these operations shall be  
 779 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 26, all operations  
 780 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

782 **Table 26 – Operations: CIM\_OpaqueManagementData**

Operation	Requirement	Messages
ModifyInstance	Mandatory	None

783 **8.11.1 CIM\_OpaqueManagementData — ModifyInstance Operation**

784 This clause details the specific requirements for the ModifyInstance operation that is applied to an  
 785 instance of CIM\_OpaqueManagementData.

786 An implementation shall support reduction of the CIM\_OpaqueManagementData.DataSize property  
 787 through the ModifyInstance operation. An implementation shall accept the ModifyInstance operation if and  
 788 only if the new value of the DataSize property is less than its current value.

789 **8.12 CIM\_OpaqueManagementDataCapabilities Operations**

790 All operations in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

792 **8.13 CIM\_StorageExtent Operations**

793 All operations in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

795 **8.14 CIM\_PhysicalMedia Operations**

796 All operations in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

798 **8.15 CIM\_PhysicalMemory Operations**

799 All operations in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

801 **8.16 CIM\_HostedService Operations**

802 Table 27 lists implementation requirements for operations. If implemented, these operations shall be  
 803 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 27, all operations  
 804 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

806 **Table 27 – Operations: CIM\_HostedService**

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

807 **8.17 CIM\_ServiceAffectsElement Operations**

808 Table 28 lists implementation requirements for operations. If implemented, these operations shall be  
 809 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 28, all operations  
 810 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

812 **Table 28 – Operations: CIM\_ServiceAffectsElement**

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

813 **8.18 CIM\_ServiceAffectsElementWithQuota Operations**

814 Table 29 lists implementation requirements for operations. If implemented, these operations shall be  
 815 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 29, all operations  
 816 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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



818 The requirements in Table 29 are in addition to those specified for the superclass  
 819 CIM\_ServiceAffectsElement in Table 28.

820 **Table 29 – Operations: CIM\_ServiceAffectsElementWithQuota**

Operation	Requirement	Messages
CreateInstance	Optional	None
ModifyInstance	Optional	None

821 **8.18.1 CIM\_ServiceAffectsElementWithQuota — CreateInstance Operation**

822 This clause details the specific requirements for the CreateInstance operation that is applied to  
 823 CIM\_ServiceAffectsElementWithQuota.

824 An implementation that supports assignment of quotas shall do so through creation of an instance of  
 825 CIM\_ServiceAffectsElementWithQuota between an instance of CIM\_Identity and the central instance.

826 **8.18.2 CIM\_ServiceAffectsElementWithQuota — ModifyInstance Operation**

827 This clause details the specific requirements for the ModifyInstance operation that is applied to an  
 828 instance of CIM\_ServiceAffectsElementWithQuota.

829 An implementation that supports assignment of quotas shall also support modification of quotas through  
 830 modification of the CIM\_ServiceAffectsElementWithQuota.Quota property. Such an implementation shall  
 831 fail any request to reduce the Quota property to a value less than the value of the QuotaUsed property in  
 832 the same instance of CIM\_ServiceAffectsElementWithQuota.

833 **8.19 CIM\_AssociatedPrivilege Operations**

834 Table 30 lists implementation requirements for operations. If implemented, these operations shall be  
 835 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 30, all operations  
 836 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

838 **Table 30 – Operations: CIM\_AssociatedPrivilege**

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

839 **8.19.1 CIM\_AssociatedPrivilege — DeleteInstance Operation**

840 This clause details the specific requirements for the DeleteInstance operation that is applied to  
 841 CIM\_AssociatedPrivilege.

842 An implementation that supports assignment of access rights through the  
 843 CIM\_OpaqueManagementDataService.AssignAccess() method shall support deletion of  
 844 CIM\_AssociatedPrivilege instances through the DeleteInstance operation.

845 **8.20 CIM\_BasedOn Operations**

846 Table 31 lists implementation requirements for operations. If implemented, these operations shall be  
 847 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 31, all operations  
 848 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

850 **Table 31 – Operations: CIM\_BasedOn**

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

851 **8.21 CIM\_ElementCapabilities Operations**

852 Table 32 lists implementation requirements for operations. If implemented, these operations shall be  
 853 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 32, all operations  
 854 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

856 **Table 32 – Operations: CIM\_ElementCapabilities**

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

857 **8.22 CIM\_RealizesExtent Operations**

858 Table 33 lists implementation requirements for operations. If implemented, these operations shall be  
 859 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 33, all operations  
 860 in the default list in 8.9 shall be implemented as defined in [DSP0200](#).

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

862 **Table 33 – Operations: CIM\_RealizesExtent**

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

## 863 9 Use Cases

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

### 867 9.1 Base Instances

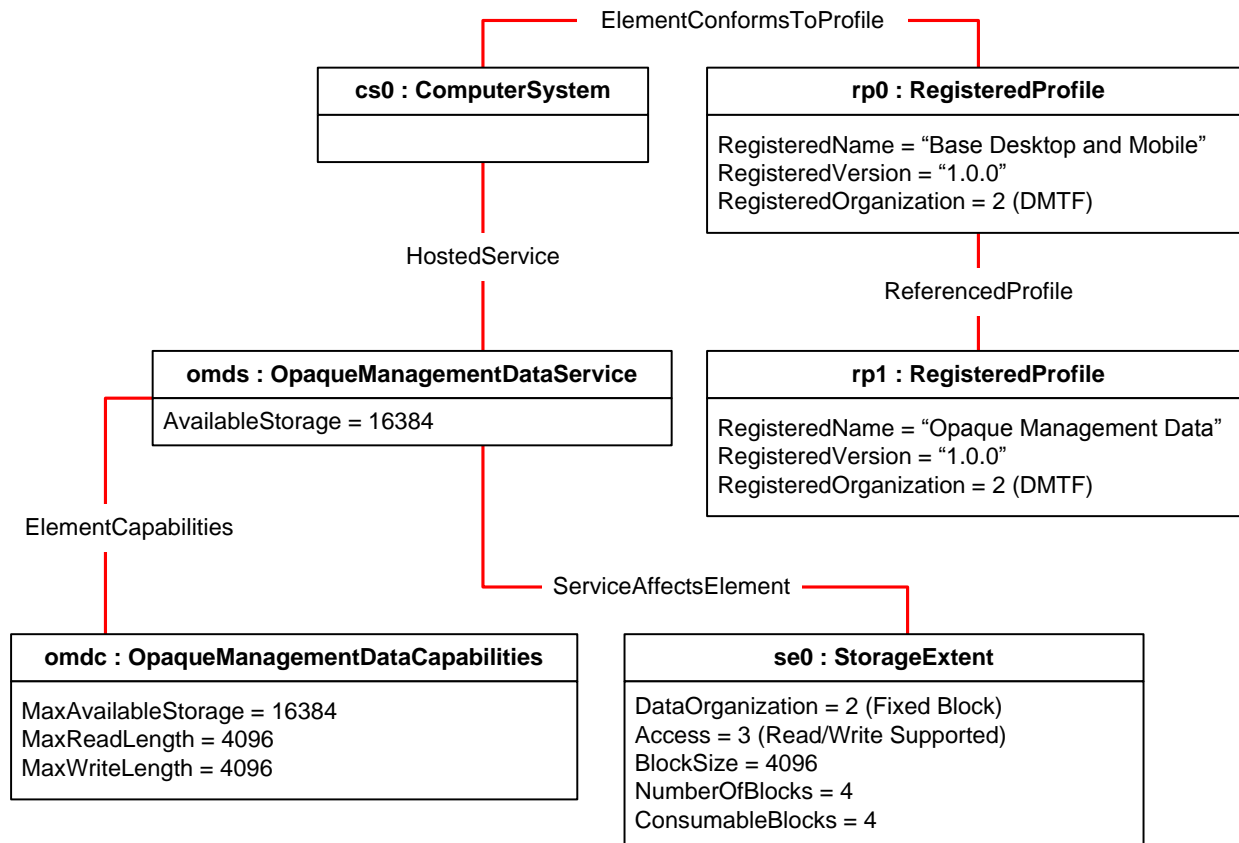
868 Figure 2 shows an example of the base instances of an implementation of the *Opaque Management Data*  
869 *Profile*. Support for the *Opaque Management Data Profile* is advertised through the  
870 CIM\_RegisteredProfile instance rp1, which is associated to the CIM\_RegisteredProfile for the [Base](#)  
871 [Desktop and Mobile Profile](#) through an instance of CIM\_ReferencedProfile.

872 An instance of the central class CIM\_OpaqueManagementDataService is scoped to the  
873 CIM\_ComputerSystem instance cs0 through an instance of CIM\_HostedService.

874 An instance of CIM\_OpaqueManagementDataCapabilities describes the capabilities of the  
875 CIM\_OpaqueManagementDataService instance to which it is associated through a  
876 CIM\_ElementCapabilities instance. In this implementation, the total number of bytes of storage (allocated  
877 + unallocated) is 16384, with zero bytes allocated to CIM\_OpaqueManagementData instances, and the  
878 maximum read and write length values are 4096. Note that  
879 CIM\_OpaqueManagementDataService.AvailableStorage cannot exceed  
880 CIM\_OpaqueManagementDataCapabilities.MaxAvailableStorage.

881 The implementation supports the optional class CIM\_StorageExtent, which describes the storage extent  
882 used for instances of CIM\_OpaqueManagementData. The CIM\_StorageExtent instance is associated  
883 with the CIM\_OpaqueManagementDataService instance by an instance of CIM\_ServiceAffectsElement.  
884 The DataOrganization property describes the organization of the data in the storage extent. In this  
885 instance the organization type is Fixed Block. The storage extent is readable and writeable, and has a  
886 BlockSize of 4096 bytes. Note that CIM\_StorageExtent.BlockSize contains the same value as  
887 CIM\_OpaqueManagementDataCapabilities.MaxReadLength and  
888 CIM\_OpaqueManagementDataCapabilities.MaxWriteLength by coincidence. For each instance of  
889 CIM\_StorageExtent there are 4 each of the number of blocks available and consumable by the user.

890 Note that in this system instance there are no instances of CIM\_Identity yet associated with the  
891 CIM\_OpaqueManagementDataService.



892

893

Figure 2 – Base Instances

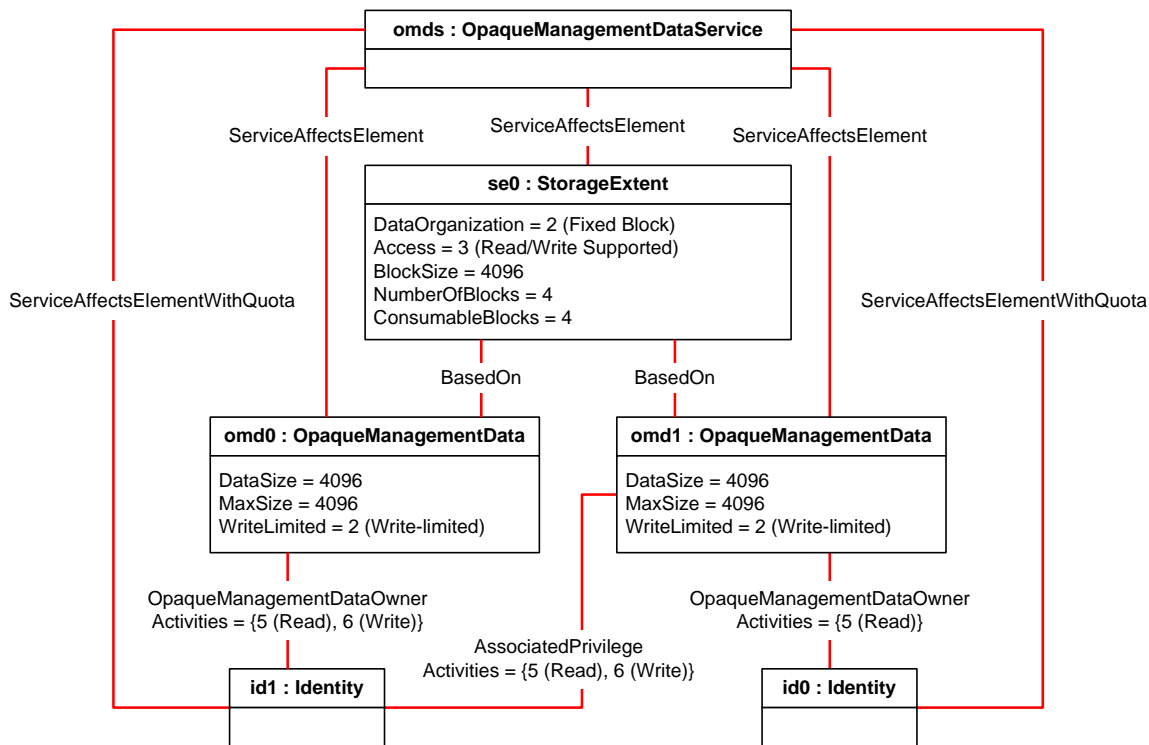
894 **9.2 CIM\_OpaqueManagementData Instances**

895 In Figure 3 the CIM\_OpaqueManagementDataService (the service) is managing two instances of  
 896 CIM\_OpaqueManagementData, each of which was created by a client using  
 897 CIM\_OpaqueManagementDataService.Create(). Each instance CIM\_OpaqueManagementData has a  
 898 write limitation, as shown by WriteLimited value = TRUE.

899 Each user is represented by an instance of CIM\_Identity, which is associated to each  
 900 CIM\_OpaqueManagementData instance to which it has access through an instance of  
 901 CIM\_AssociatedPrivilege.

902 CIM\_Identity instance id1 is the owner of CIM\_OpaqueManagementData instance omd0 by virtue of  
 903 having created omd0 using CIM\_OpaqueManagementDataService.Create(). Likewise, CIM\_Identity  
 904 instance id0 is the owner of CIM\_OpaqueManagementData instance omd1. CIM\_Identity instance id0 has  
 905 granted id1 read and write access to omd1.

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



912

913

Figure 3 – CIM\_OpaqueManagementData Instances

914 **9.3 Physical Modeling**

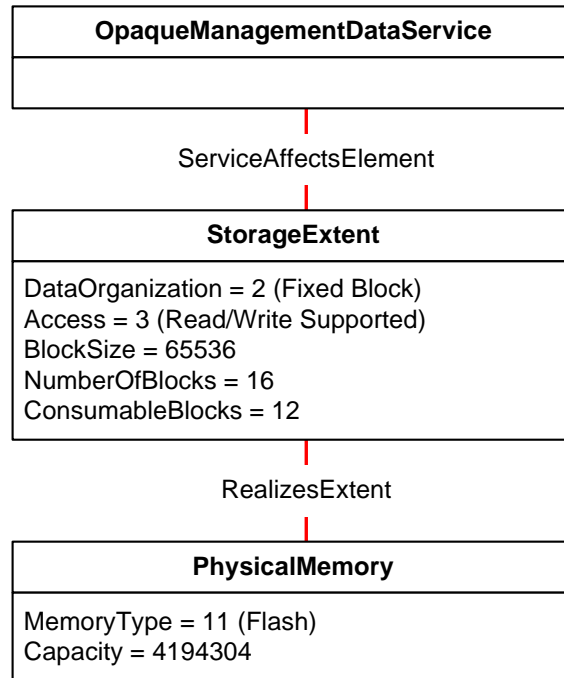
915 Figure 4 depicts an example of the physical aspects of an *Opaque Management Data Profile*  
 916 implementation. The central class CIM\_OpaqueManagementDataService is instantiated.

917 The instance of CIM\_StorageExtent associated with the CIM\_OpaqueManagementDataService instance  
 918 through CIM\_ServiceAffectsElement describes the logical characteristics of the storage used by the  
 919 service to store opaque management data:

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

923 The CIM\_PhysicalMemory instance associated with the CIM\_StorageExtent instance through  
 924 CIM\_Realizes describes the physical aspects of this storage:

- 925 • It is a flash device.
- 926 • Its capacity is 4 MB. Since this is considerably larger than the associated CIM\_StorageExtent, it  
 927 could contain more than just opaque management data.



928

929

**Figure 4 – Physical Modeling**

930 **9.4 Create an Instance of CIM\_OpaqueManagementData**

931 A client can create an instance of CIM\_OpaqueManagementData as follows:

- 932 1) Find the CIM\_OpaqueManagementDataService and verify it is enabled as follows:
- 933 a) Starting at the CIM\_ComputerSystem instance for the managed system, look for instances
- 934 of CIM\_OpaqueManagementDataService associated through CIM\_HostedService. If
- 935 multiple instances of CIM\_OpaqueManagementDataService are found, select the one of
- 936 interest using the property values of each instance.
- 937 b) Verify that CIM\_OpaqueManagementDataService.EnabledState property value has the
- 938 value 2 (Enabled). If the value is 3, then the service is Disabled. If the value is 0
- 939 (Unknown), then the service is in transition. For this example, assume EnabledState
- 940 contains 2 (Enabled).
- 941 2) Determine the user identity that will have access to CIM\_OpaqueManagementData by finding
- 942 the instance of CIM\_Identity you will use that is associated with
- 943 CIM\_OpaqueManagementDataService through an instance of the CIM\_ServiceAffectsElement
- 944 association.
- 945 3) Create an instance of CIM\_OpaqueManagementData by invoking
- 946 CIM\_OpaqueManagementService.Create() with required parameters. If
- 947 CIM\_OpaqueManagementDataService.Create() completes successfully, the new
- 948 CIM\_OpaqueManagementData instance will be associated with the owning CIM\_Identity
- 949 instance through a new instance of CIM\_OpaqueManagementDataOwner.

950 **9.5 Read Data Represented by an Instance of CIM\_OpaqueManagementData**

951 A client can use CIM\_OpaqueManagementDataService.Read() or

952 CIM\_OpaqueManagementDataService.ExportToURI() to read from an instance of

953 CIM\_OpaqueManagementData.

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

958 Note that CIM\_OpaqueManagementDataService.ExportToURI() will read from the storage represented by  
959 the CIM\_OpaqueManagementData instance and write to the URI supplied.

## 960 **9.6 Write Data Represented by an Instance of CIM\_OpaqueManagementData**

961 A client can use CIM\_OpaqueManagementDataService.Write() or  
962 CIM\_OpaqueManagementDataService.ImportFromURI() to write to an instance of  
963 CIM\_OpaqueManagementData.

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

968 Note that CIM\_OpaqueManagementDataService.ImportFromURI() will read from the URI supplied and  
969 write to the storage represented by the CIM\_OpaqueManagementData instance.

## 970 **9.7 Overwrite and Truncate Data Represented by an Instance of 971 CIM\_OpaqueManagementData**

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

## 976 **9.8 Delete an Instance of CIM\_OpaqueManagementData**

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

## 979 **9.9 Enumerate Instances of CIM\_OpaqueManagementData**

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

## 981 **9.10 Manage CIM\_OpaqueManagementData Access**

982 If the CIM\_OpaqueManagementDataCapabilities.SupportedMethods property contains the value 9  
983 (AssignAccess) then the CIM\_OpaqueManagementData.AssignAccess() method can be used to manage  
984 access to CIM\_OpaqueManagementData instances.

985 **10 CIM Elements**

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

989 **Table 34 – CIM Elements: Opaque Management Data Profile**

Element Name	Requirement	Description
<b>Classes</b>		
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.
<b>Indications</b>		
None defined in this profile		

990 **10.1 CIM\_OpaqueManagementDataService**

991 Table 35 contains requirements for elements of CIM\_OpaqueManagementDataService. These  
 992 requirements are in addition to those specified for the superclass CIM\_EnabledLogicalElement in the  
 993 [Enabled Logical Element Profile](#).

994 **Table 35 – Class: CIM\_OpaqueManagementDataService**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
Name	Mandatory	<b>Key</b>
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.

995 **10.2 CIM\_OpaqueManagementData**

996 Table 36 contains requirements for elements of CIM\_OpaqueManagementData. The requirements in  
 997 Table 36 are in addition to those specified for the superclass CIM\_StorageExtent in Table 38.

998 **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.

999 **10.3 CIM\_OpaqueManagementDataCapabilities**

1000 Table 37 contains requirements for elements of CIM\_OpaqueManagementDataCapabilities. These  
 1001 requirements are in addition to those specified for the superclass  
 1002 CIM\_EnabledLogicalElementCapabilities in the [Enabled Logical Element Profile](#).

1003 **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.

1004 **10.4 CIM\_StorageExtent**

1005 Table 38 contains requirements for elements of CIM\_StorageExtent. These requirements are in addition  
 1006 to those specified for superclass CIM\_EnabledLogicalElement in the [Enabled Logical Element Profile](#).

1007 **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	<b>Key</b>
SystemName	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
DeviceID	Mandatory	<b>Key</b>

1008 **10.5 CIM\_PhysicalMedia**

1009 Table 39 contains the requirements for elements of CIM\_PhysicalMedia.

1010 **Table 39 – Class: CIM\_PhysicalMedia**

Elements	Requirement	Notes
Tag	Mandatory	<b>Key</b>
CreationClassName	Mandatory	<b>Key</b>
Capacity	Mandatory	None
MediaType	Mandatory	None
Manufacturer	Conditional	See the <a href="#">Physical Asset Profile</a> .
Model	Conditional	See the <a href="#">Physical Asset Profile</a> .
Serial Number	Conditional	See the <a href="#">Physical Asset Profile</a> .
PartNumber	Conditional	See the <a href="#">Physical Asset Profile</a> .
SKU	Conditional	See the <a href="#">Physical Asset Profile</a> .
CanBeFRUed	Optional	Should be implemented when the PhysicalMedia can be field replaced.
ElementName	Mandatory	Matches pattern “.*”.

1011 **10.6 CIM\_PhysicalMemory**

1012 An implementation that supports CIM\_PhysicalMemory as specified in clause 7.3.11 shall support the  
 1013 CIM\_PhysicalMemory element requirements defined in the [Physical Asset Profile](#).

1014 **10.7 CIM\_RegisteredProfile**

1015 An implementation shall support CIM\_RegisteredProfile. With the exception of the mandatory values  
 1016 specified for the properties in Table 40, the behavior of the CIM\_RegisteredProfile instance is defined by  
 1017 the [Profile Registration Profile](#).

1018 **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).

1019 **10.8 CIM\_HostedService**

1020 Table 41 contains the requirements for elements of CIM\_HostedService.

1021 **Table 41 – Class: CIM\_HostedService**

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

1022 **10.9 CIM\_ServiceAffectsElement (CIM\_OpaqueManagementData)**

1023 Table 42 contains the element requirements for CIM\_ServiceAffectsElement between  
 1024 CIM\_OpaqueManagementDataService and CIM\_OpaqueManagementData.

1025 **Table 42 – Class: CIM\_ServiceAffectsElement**

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

1026 **10.10 CIM\_ServiceAffectsElement (CIM\_StorageExtent)**

1027 Table 43 contains the element requirements for CIM\_ServiceAffectsElement between  
 1028 CIM\_OpaqueManagementDataService and CIM\_StorageExtent.

1029 **Table 43 – Class: CIM\_ServiceAffectsElement**

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

1030 **10.11 CIM\_ServiceAffectsElement (CIM\_Identity)**

1031 Table 44 contains the element requirements for CIM\_ServiceAffectsElement between  
 1032 CIM\_OpaqueManagementDataService and CIM\_Identity.

1033 **Table 44 – Class: CIM\_ServiceAffectsElement**

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

1034 **10.12 CIM\_ServiceAffectsElementWithQuota**

1035 Table 45 contains the requirements for elements of CIM\_ServiceAffectsElementWithQuota. The  
 1036 requirements in Table 45 are in addition to those specified for the superclass CIM\_ServiceAffectsElement  
 1037 in Table 42.

1038 **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.

1039 **10.13 CIM\_AssociatedPrivilege**

1040 Table 46 contains the requirements for elements of CIM\_AssociatedPrivilege.

1041 **Table 46 – Class: CIM\_AssociatedPrivilege**

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

1042 **10.14 CIM\_OpaqueManagementDataOwner**

1043 No properties are required beyond those required for CIM\_AssociatedPrivilege in 10.13.

1044 **10.15 CIM\_BasedOn**

1045 Table 47 contains the requirements for elements of CIM\_BasedOn.

1046 **Table 47 – Class: CIM\_BasedOn**

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

1047 **10.16 CIM\_ElementCapabilities**

1048 Table 48 contains the requirements for elements of CIM\_ElementCapabilities.

1049 **Table 48 – Class: CIM\_ElementCapabilities**

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

1050 **10.17 CIM\_RealizesExtent**

1051 Table 49 contains the requirements for elements of CIM\_RealizesExtent.

1052 **Table 49 – Class: CIM\_RealizesExtent**

Elements	Requirement	Notes
Antecedent	Mandatory	<b>Key:</b> This property shall contain a reference to an instance of CIM_PhysicalMedia or CIM_PhysicalMemory. Cardinality 0..1
Dependent	Mandatory	<b>Key:</b> 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	2009-05-20	
1.0.1	2009-10-21	Corrections to Figure 1
1.0.2	2014-03-24	Corrected standards messages for methods. Clarified an error case for Write() method.

1053  
1054  
1055  
1056  
1057

1058