



1

2

3

4

Document Number: DSP1070

Date: 2009-06-10

Version: 1.0.0

5 **Opaque Management Data Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9

10 Copyright Notice

11 Copyright © 2008, 2009 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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().....	28
55	8.7 CIM_OpaqueManagementDataService.ReassignOwnership()	29
56	8.8 CIM_OpaqueManagementDataService.AssignAccess().....	30
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	32
61	8.13 CIM_StorageExtent Operations.....	32
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	33
66	8.18 CIM_ServiceAffectsElementWithQuota Operations	33
67	8.19 CIM_AssociatedPrivilege Operations	34
68	8.20 CIM_BasedOn Operations.....	34
69	8.21 CIM_ElementCapabilities Operations.....	34
70	8.22 CIM_RealizesExtent Operations.....	35
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	
124	Values	26
125	Table 15 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Standard	
126	Messages.....	27
127	Table 16 – CIM_OpaqueManagementDataService.ImportFromURI() Method: Parameters.....	27
128	Table 17 – CIM_OpaqueManagementDataService.Lock() Method: Return Code Values	28
129	Table 18 – CIM_OpaqueManagementDataService.Lock() Method: Standard Messages	28
130	Table 19 – CIM_OpaqueManagementDataService.Lock() Method: Parameters	28
131	Table 20 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Return Code	
132	Values	29

133 Table 21 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Standard
 134 Messages 29
 135 Table 22 – CIM_OpaqueManagementDataService.ReassignOwnership() Method: Parameters 29
 136 Table 23 – CIM_OpaqueManagementDataService.AssignAccess() Method: Return Code Values 30
 137 Table 24 – CIM_OpaqueManagementDataService.AssignAccess() Method: Standard Messages 30
 138 Table 25 – CIM_OpaqueManagementDataService.AssignAccess() Method: Parameters 31
 139 Table 26 – Operations: CIM_OpaqueManagementData 32
 140 Table 27 – Operations: CIM_HostedService 32
 141 Table 28 – Operations: CIM_ServiceAffectsElement 33
 142 Table 29 – Operations: CIM_ServiceAffectsElementWithQuota 33
 143 Table 30 – Operations: CIM_AssociatedPrivilege 34
 144 Table 31 – Operations: CIM_BasedOn 34
 145 Table 32 – Operations: CIM_ElementCapabilities 35
 146 Table 33 – Operations: CIM_RealizesExtent 35
 147 Table 34 – CIM Elements: Opaque Management Data Profile 40
 148 Table 35 – Class: CIM_OpaqueManagementDataService 40
 149 Table 36 – Class: CIM_OpaqueManagementData 41
 150 Table 37 – Class: CIM_OpaqueManagementDataCapabilities 41
 151 Table 38 – Class: CIM_StorageExtent 42
 152 Table 39 – Class: CIM_PhysicalMedia 42
 153 Table 40 – Class: CIM_RegisteredProfile 43
 154 Table 41 – Class: CIM_HostedService 43
 155 Table 42 – Class: CIM_ServiceAffectsElement 43
 156 Table 43 – Class: CIM_ServiceAffectsElement 44
 157 Table 44 – Class: CIM_ServiceAffectsElement 44
 158 Table 45 – Class: CIM_ServiceAffectsElementWithQuota 44
 159 Table 46 – Class: CIM_AssociatedPrivilege 45
 160 Table 47 – Class: CIM_BasedOn 45
 161 Table 48 – Class: CIM_ElementCapabilities 45
 162 Table 49 – Class: CIM_RealizesExtent 46
 163

165

Foreword

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

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

170 **Acknowledgments**

171 The authors wish to acknowledge the following people.

172 Editor:

- 173 • David Hines – Intel

174 Contributors:

- 175 • Murali Rajagopal
- 176 • Hemal Shah – Broadcom
- 177 • Sharon L Smith – Intel

178 Participants from the DMTF Desktop and Mobile Working Group:

- 179 • Bob Blair – AMD
- 180 • Jon Hass – Dell
- 181 • Jeff Hilland – HP
- 182 • Joel Clark – Intel
- 183 • Steve Hand – Symantec

184

185

Introduction

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

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

192

Opaque Management Data Profile

193 1 Scope

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

196 2 Normative References

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

200 2.1 Approved References

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

203 DMTF DSP0200, *CIM Operations over HTTP 1.2*,
204 http://www.dmtf.org/standards/published_documents/DSP200.pdf

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

207 DMTF DSP1011, *Physical Asset Profile 1.0*,
208 http://www.dmtf.org/standards/published_documents/DSP1011_1.0.1.pdf

209 DMTF DSP1033, *Profile Registration Profile 1.0*,
210 http://www.dmtf.org/standards/published_documents/DSP1033_1.0.0.pdf

211 DMTF DSP1034, *Simple Identity Management Profile 1.0*,
212 http://www.dmtf.org/standards/published_documents/DSP1034_1.0.0.pdf

213 DMTF DSP1058, *Base Desktop and Mobile Profile 1.0*,
214 http://www.dmtf.org/standards/published_documents/DSP1058_1.0.0.pdf

215 DMTF DSP1080, *Enabled Logical Element Profile 1.0*,
216 http://www.dmtf.org/standards/published_documents/DSP1080_1.0.0.pdf

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

219 2.2 Other References

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

222 3 Terms and Definitions

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

- 225 **3.1**
- 226 **can**
- 227 used for statements of possibility and capability, whether material, physical, or causal
- 228 **3.2**
- 229 **cannot**
- 230 used for statements of possibility and capability, whether material, physical, or causal
- 231 **3.3**
- 232 **conditional**
- 233 indicates requirements to be followed strictly in order to conform to the document when the specified
- 234 conditions are met
- 235 **3.4**
- 236 **mandatory**
- 237 indicates requirements to be followed strictly in order to conform to the document and from which no
- 238 deviation is permitted
- 239 **3.5**
- 240 **may**
- 241 indicates a course of action permissible within the limits of the document
- 242 **3.6**
- 243 **need not**
- 244 indicates a course of action permissible within the limits of the document
- 245 **3.7**
- 246 **optional**
- 247 indicates a course of action permissible within the limits of the document
- 248 **3.8**
- 249 **referencing profile**
- 250 indicates a profile that owns the definition of this class and can include a reference to this profile in its
- 251 "Related Profiles" table
- 252 **3.9**
- 253 **shall**
- 254 indicates requirements to be followed strictly in order to conform to the document and from which no
- 255 deviation is permitted
- 256 **3.10**
- 257 **shall not**
- 258 indicates requirements to be followed in order to conform to the document and from which no deviation is
- 259 permitted
- 260 **3.11**
- 261 **should**
- 262 indicates that among several possibilities, one is recommended as particularly suitable, without
- 263 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 264 **3.12**
- 265 **should not**
- 266 indicates that a certain possibility or course of action is deprecated but not prohibited

267 **3.13**
 268 **unspecified**
 269 indicates that this profile does not define any constraints for the referenced CIM element or operation

270 **4 Symbols and Abbreviated Terms**

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

272 **4.1**
 273 **CIM**
 274 Common Information Model

275 **5 Synopsis**

276 **Profile Name:** Opaque Management Data

277 **Version:** 1.0.0

278 **Organization:** DMTF

279 **CIM Schema Version:** 2.22

280 **Central Class:** CIM_OpaqueManagementDataService

281 **Scoping Class:** CIM_ComputerSystem

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

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

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

290 **Table 1 – Related Profiles**

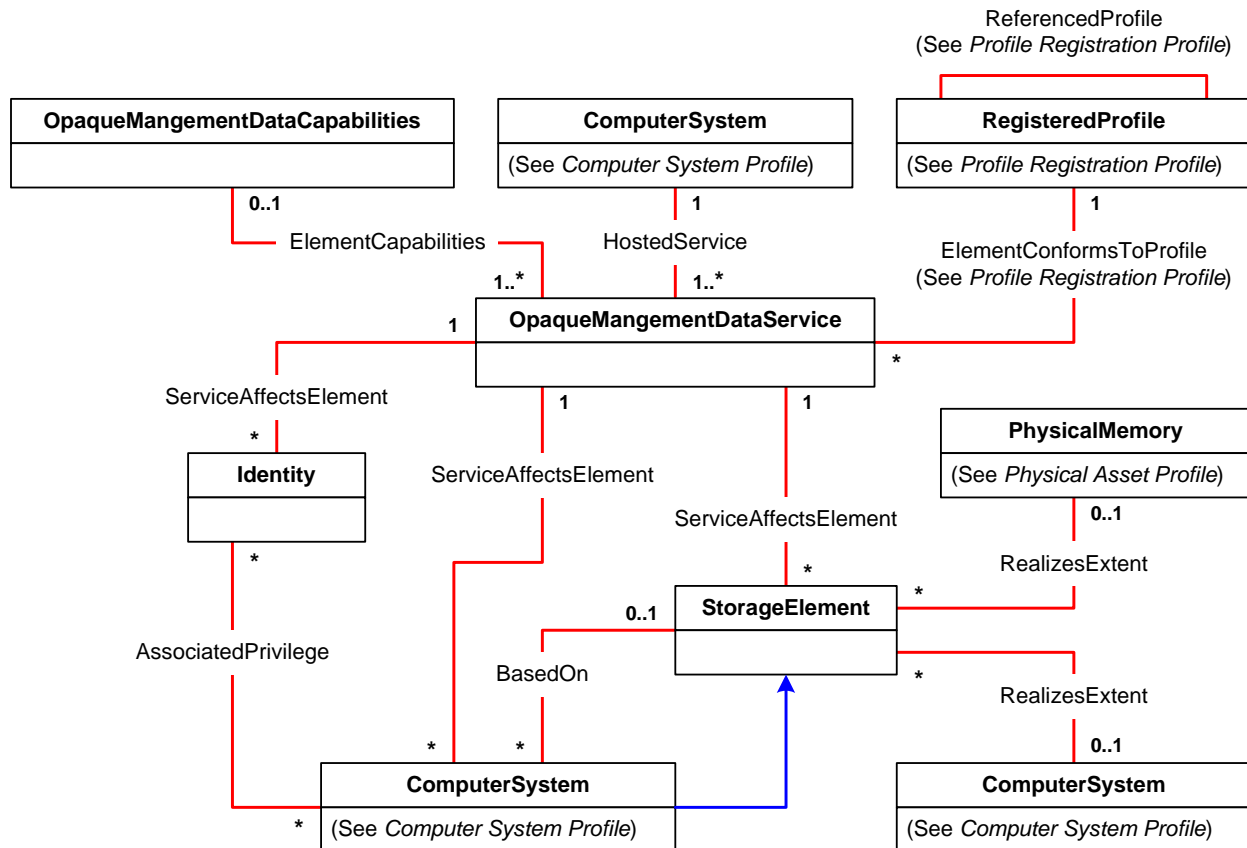
Profile Name	Organization	Version	Requirement
Profile Registration	DMTF	1.0	Mandatory
Simple Identity Management	DMTF	1.0	Mandatory
Physical Asset	DMTF	1.0.	Optional
Enabled Logical Element	DMTF	1.0	Mandatory

291 **6 Description**

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

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

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



304

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

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

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

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

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

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

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

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

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

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

333 7 Implementation Requirements

334 This clause specifies the requirements related to the arrangement of instances and properties for
335 implementations of this profile. Methods are specified in clause 8, and properties are specified in clause
336 10.

337 7.1 CIM_OpaqueManagementDataService

338 At least one instance of CIM_OpaqueManagementDataService shall be supported.

339 7.1.1 CIM_OpaqueManagementDataService.LockTimeout

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

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

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

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

351 7.1.2 CIM_OpaqueManagementDataService.MaxCreateSize

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

356 **7.1.3 CIM_OpaqueManagementDataService.AvailableStorage**

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

362 CIM_OpaqueManagementDataService.AvailableStorage shall not exceed
363 CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage.

364 **7.1.4 Managing CIM_OpaqueManagementDataService.ElementName**

365 An implementation may support management of the ElementName property of
366 CIM_OpaqueManagementDataService. The abstract [Enabled Logical Element Profile](#) specifies
367 requirements for supporting ElementName management in subclasses of CIM_EnabledLogicalElement.
368 The implementation of CIM_OpaqueManagementDataService shall meet the requirements in the [Enabled](#)
369 [Logical Element Profile](#), with CIM_OpaqueManagementDataService in place of
370 CIM_EnabledLogicalElement and CIM_OpaqueManagementDataCapabilities in place of
371 CIM_EnabledLogicalElementCapabilities.

372 **7.1.5 Managing CIM_OpaqueManagementDataService State**

373 An implementation may support management of CIM_OpaqueManagementDataService state. The
374 abstract [Enabled Logical Element Profile](#) specifies requirements for supporting state management in
375 subclasses of CIM_EnabledLogicalElement. The implementation of
376 CIM_OpaqueManagementDataService shall meet the requirements of the [Enabled Logical Element](#)
377 [Profile](#), with CIM_OpaqueManagementDataService in place of CIM_EnabledLogicalElement and
378 CIM_OpaqueManagementDataCapabilities in place of CIM_EnabledLogicalElementCapabilities.

379 **7.1.6 CIM_HostedService**

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

382 **7.2 CIM_OpaqueManagementDataCapabilities**

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

385 **7.2.1 CIM_OpaqueManagementDataCapabilities.MaxAvailableStorage**

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

391 **7.2.2 CIM_OpaqueManagementDataCapabilities.MaxReadLength**

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

399 **7.2.3 CIM_OpaqueManagementDataCapabilities.MaxWriteLength**

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

407 **7.2.4 CIM_OpaqueManagementDataCapabilities.SupportedMethods**

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

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

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

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

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

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

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

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

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

432 **7.2.5 CIM_OpaqueManagementDataCapabilities.SupportedExportURISchemes**

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

436 **7.2.6 CIM_OpaqueManagementDataCapabilities.SupportedImportURISchemes**

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

440 **7.3 Representing Opaque Management Data**

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

444 **7.3.1 Managing CIM_OpaqueManagementData.ElementName**

445 An implementation may support management of the ElementName property of
446 CIM_OpaqueManagementData. The abstract [Enabled Logical Element Profile](#) specifies requirements for
447 supporting ElementName management in subclasses of CIM_EnabledLogicalElement. The
448 implementation of CIM_OpaqueManagementData shall meet the requirements in the [Enabled Logical
449 Element Profile](#), with CIM_OpaqueManagementData in place of CIM_EnabledLogicalElement.

450 **7.3.2 Managing CIM_OpaqueManagementData State**

451 An implementation may support management of CIM_OpaqueManagementData state. The abstract
452 [Enabled Logical Element Profile](#) specifies requirements for supporting state management in subclasses
453 of CIM_EnabledLogicalElement. The implementation of CIM_OpaqueManagementData shall meet the
454 requirements of the [Enabled Logical Element Profile](#), with CIM_OpaqueManagementData in place of
455 CIM_EnabledLogicalElement.

456 **7.3.3 Ownership of Opaque Management Data**

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

461 **7.3.3.1 CIM_OpaqueManagementDataOwner**

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

467 **7.3.3.2 Reassignment of Ownership**

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

470 **7.3.3.2.1 Reassignment of Ownership Is Supported**

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

474 **7.3.3.2.1.1 CIM_OpaqueManagementDataService.ReassignOwnership()**

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

477 **7.3.3.2.1.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods**

478 The CIM_OpaqueManagementDataCapabilities.SupportedMethods array property shall contain the value
479 8 (ReassignOwnership).

480 **7.3.3.2.2 Reassignment of Ownership Is Not Supported**

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

484 **7.3.3.2.2.1 CIM_OpaqueManagementDataService.ReassignOwnership()**

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

486 **7.3.3.2.2.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods**

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

489 **7.3.4 Access Rights to Opaque Management Data**

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

495 **7.3.4.1 CIM_AssociatedPrivilege**

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

501 **7.3.4.1.1 CIM_AssociatedPrivilege.Activities**

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

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

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

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

514 **7.3.4.1.2 CIM_AssociatedPrivilege.ActivityQualifiers**

515 The value “DMTF: DSP1070:” 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 **7.3.4.1.3 CIM_AssociatedPrivilege.QualifierFormats**

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

520 7.3.4.2 Assignment of Access Rights

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

524 7.3.4.2.1 Assignment of Access Rights Is Supported

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

528 7.3.4.2.1.1 CIM_OpaqueManagementDataService.AssignAccess()

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

531 7.3.4.2.1.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods

532 The CIM_OpaqueManagementDataCapabilities.SupportedMethods array property shall contain the value
533 9 (AssignAccess).

534 7.3.4.2.2 Assignment of Access Rights Is Not Supported

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

538 7.3.4.2.2.1 CIM_OpaqueManagementDataService.AssignAccess()

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

540 7.3.4.2.2.2 CIM_OpaqueManagementDataCapabilities.SupportedMethods

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

543 7.3.5 CIM_OpaqueManagementData.DataSize

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

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

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

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

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

554 7.3.6 CIM_OpaqueManagementData.MaxSize

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

557 **7.3.7 CIM_OpaqueManagementData.DataFormat**

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

560 **7.3.8 CIM_OpaqueManagementData.LastAccessed**

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

564 **7.3.9 CIM_OpaqueManagementData.WriteLimited**

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

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

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

574 **7.3.10 Representing the Storage of Opaque Management Data**

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

579 **7.3.11 Representing the Physical Realization of Opaque Management Data**

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

585 **7.3.12 Representing Management of Opaque Management Data**

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

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

594 **7.3.13 Representing Management of Users of Opaque Management Data**

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

597 7.3.13.1 CIM_ServiceAffectsElement

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

602 7.3.13.2 Representing Opaque Management Data Storage Quotas

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

607 7.3.13.2.1 CIM_ServiceAffectsElementWithQuota

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

611 7.3.13.2.1.1 CIM_ServiceAffectsElementWithQuota.Quota

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

615 7.3.13.2.1.2 CIM_ServiceAffectsElementWithQuota.QuotaUsed

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

619 7.3.13.2.1.3 CIM_ServiceAffectsElementWithQuota.QuotaUnits

620 CIM_ServiceAffectsElementWithQuota.QuotaUnits shall contain "byte".

621 7.3.14 Locking Opaque Management Data

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

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

628 7.3.14.1 Locking Opaque Management Data Is Supported

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

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

633 7.3.14.2 Locking Opaque Management Data Is Not Supported

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

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

638 **8 Methods**

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

641 **8.1 CIM_OpaqueManagementDataService.Create()**

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

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

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

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

654 **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

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

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

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

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

659 **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)

660 **8.2 CIM_OpaqueManagementDataService.Read()**

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

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

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

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

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

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

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

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

670 **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.

671 **8.3 CIM_OpaqueManagementDataService.ExportToURI()**

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

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

677 **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

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

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

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

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

683 **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).

684 **8.4 CIM_OpaqueManagementDataService.Write()**

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

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

689 **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

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

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

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

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

694 **Table 13 – CIM_OpaqueManagementDataService.Write() Method: Parameters**

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

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

695 **8.5 CIM_OpaqueManagementDataService.ImportFromURI()**

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

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

701 **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

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

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

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

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

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

Qualifiers	Name	Type	Description/Values
Required, IN	OpaqueManagementData	CIM_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)

708 **8.6 CIM_OpaqueManagementDataService.Lock()**

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

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

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

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

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

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

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

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

722 **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.

723 **8.7 CIM_OpaqueManagementDataService.ReassignOwnership()**

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

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

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

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

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

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

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

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

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

738 **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.

739 **8.8 CIM_OpaqueManagementDataService.AssignAccess()**

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

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

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

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

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

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

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

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

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

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

765 **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).

766 **8.9 Profile Conventions for Operations**

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

769 The default list of operations is as follows:

- 770 • GetInstance
- 771 • Associators
- 772 • AssociatorNames
- 773 • References
- 774 • ReferenceNames
- 775 • EnumerateInstances
- 776 • EnumerateInstanceNames

777 **8.10 CIM_OpaqueManagementDataService Operations**

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

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

780 **8.11 CIM_OpaqueManagementData Operations**

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

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

785

Table 26 – Operations: CIM_OpaqueManagementData

Operation	Requirement	Messages
ModifyInstance	Mandatory	None

786 **8.11.1 CIM_OpaqueManagementData — ModifyInstance Operation**

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

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

792 **8.12 CIM_OpaqueManagementDataCapabilities 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 **8.13 CIM_StorageExtent 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 **8.14 CIM_PhysicalMedia 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 **8.15 CIM_PhysicalMemory Operations**

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

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

804 **8.16 CIM_HostedService Operations**

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

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

809

Table 27 – Operations: CIM_HostedService

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

810 **8.17 CIM_ServiceAffectsElement Operations**

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

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

815 **Table 28 – Operations: CIM_ServiceAffectsElement**

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

816 **8.18 CIM_ServiceAffectsElementWithQuota Operations**

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

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

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

823 **Table 29 – Operations: CIM_ServiceAffectsElementWithQuota**

Operation	Requirement	Messages
CreateInstance	Optional	None
ModifyInstance	Optional	None

824 **8.18.1 CIM_ServiceAffectsElementWithQuota — CreateInstance Operation**

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

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

829 **8.18.2 CIM_ServiceAffectsElementWithQuota — ModifyInstance Operation**

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

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

836 **8.19 CIM_AssociatedPrivilege Operations**

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

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

841 **Table 30 – Operations: CIM_AssociatedPrivilege**

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

842 **8.19.1 CIM_AssociatedPrivilege — DeleteInstance Operation**

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

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

848 **8.20 CIM_BasedOn Operations**

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

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

853 **Table 31 – Operations: CIM_BasedOn**

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

854 **8.21 CIM_ElementCapabilities Operations**

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

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

859

Table 32 – Operations: CIM_ElementCapabilities

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

860 **8.22 CIM_RealizesExtent Operations**

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

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

865

Table 33 – Operations: CIM_RealizesExtent

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

866 **9 Use Cases**

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

870 **9.1 Base Instances**

871 Figure 2 shows an example of the base instances of an implementation of the *Opaque Management Data*
 872 *Profile*. Support for the *Opaque Management Data Profile* is advertised through the
 873 CIM_RegisteredProfile instance rp1, which is associated to the CIM_RegisteredProfile for the [Base](#)
 874 [Desktop and Mobile Profile](#) through an instance of CIM_ReferencedProfile.

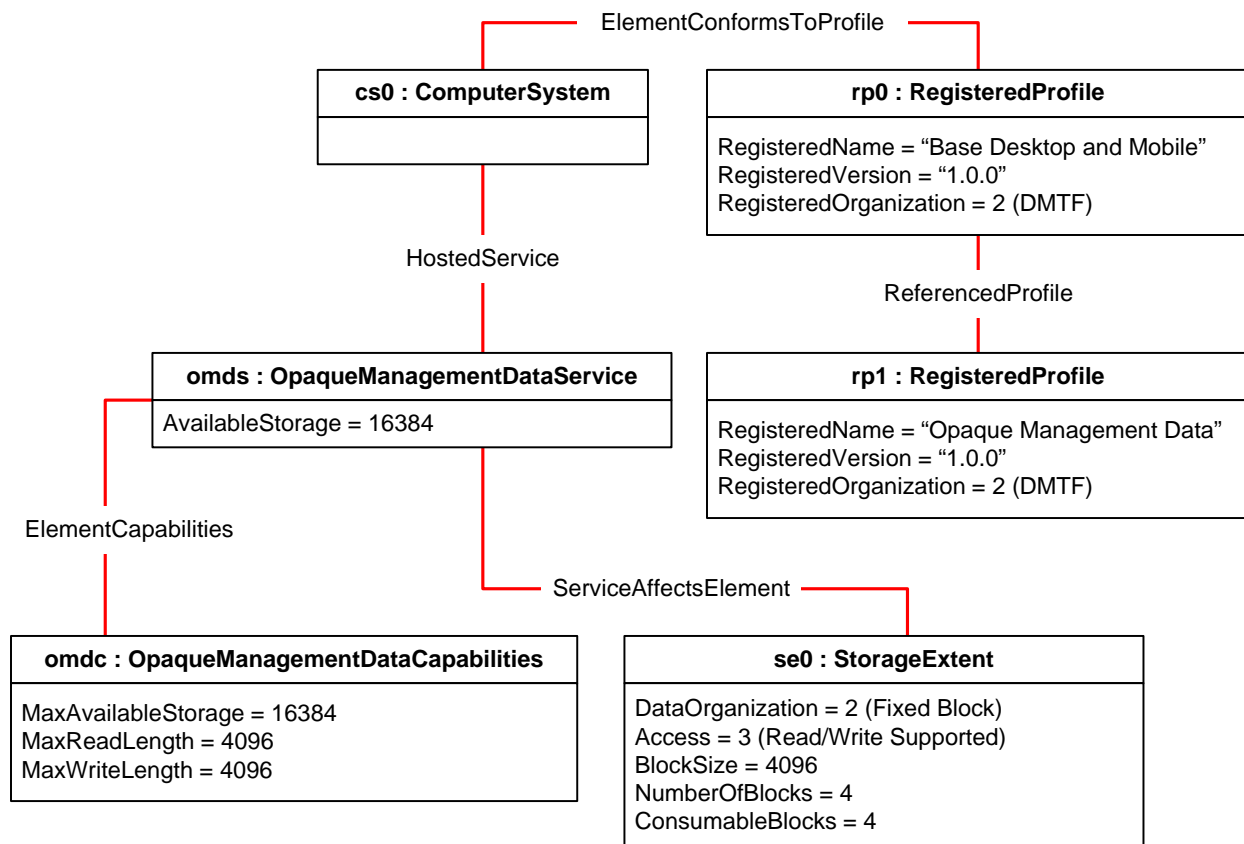
875 An instance of the central class CIM_OpaqueManagementDataService is scoped to the
 876 CIM_ComputerSystem instance cs0 through an instance of CIM_HostedService.

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

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

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

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



895

896

Figure 2 – Base Instances

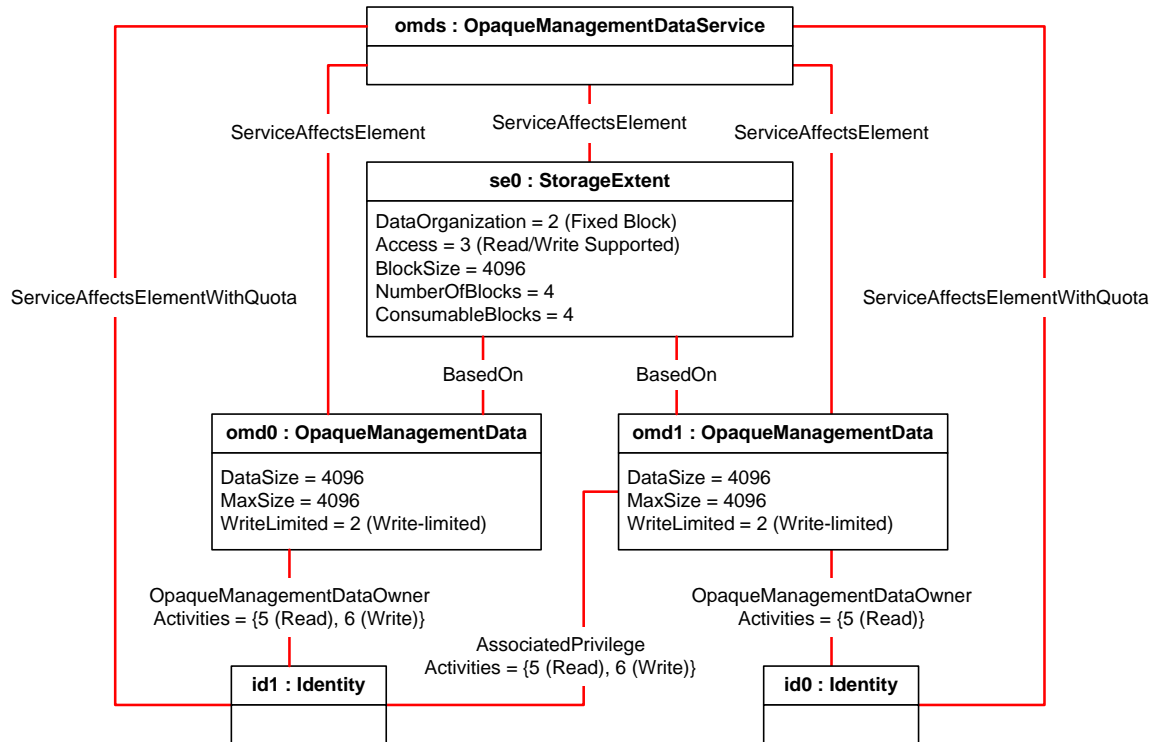
897 9.2 CIM_OpaqueManagementData Instances

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

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

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

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



915

916

Figure 3 – CIM_OpaqueManagementData Instances

917 **9.3 Physical Modeling**

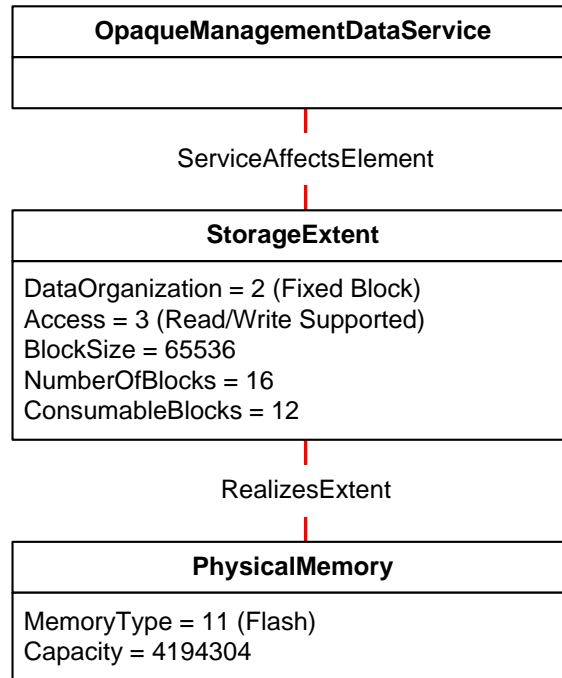
918 Figure 4 depicts an example of the physical aspects of an *Opaque Management Data Profile*
 919 implementation. The central class CIM_OpaqueManagementDataService is instantiated.

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

- 923
- It has a fixed block size of 64 KB.
 - 924 • It supports read/write access.
 - 925 • It consists of 16 blocks, 12 of which are consumable by the service.

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

- 928
- It is a flash device.
 - 929 • Its capacity is 4 MB. Since this is considerably larger than the associated CIM_StorageExtent, it
 930 could contain more than just opaque management data.



931

932

Figure 4 – Physical Modeling

933 **9.4 Create an Instance of CIM_OpaqueManagementData**

934 A client can create an instance of CIM_OpaqueManagementData as follows:

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

953 **9.5 Read Data Represented by an Instance of CIM_OpaqueManagementData**

954 A client can use CIM_OpaqueManagementDataService.Read() or

955 CIM_OpaqueManagementDataService.ExportToURI() to read from an instance of

956 CIM_OpaqueManagementData.

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

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

963 **9.6 Write Data Represented by an Instance of `CIM_OpaqueManagementData`**

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

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

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

973 **9.7 Overwrite and Truncate Data Represented by an Instance of 974 `CIM_OpaqueManagementData`**

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

979 **9.8 Delete an Instance of `CIM_OpaqueManagementData`**

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

982 **9.9 Enumerate Instances of `CIM_OpaqueManagementData`**

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

984 **9.10 Manage `CIM_OpaqueManagementData` Access**

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

988 **10 CIM Elements**

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

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

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

993 **10.1 CIM_OpaqueManagementDataService**

994 Table 35 contains requirements for elements of CIM_OpaqueManagementDataService. These
 995 requirements are in addition to those specified for the superclass CIM_EnabledLogicalElement in the
 996 [Enabled Logical Element Profile](#).

997 **Table 35 – Class: CIM_OpaqueManagementDataService**

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

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

998 **10.2 CIM_OpaqueManagementData**

999 Table 36 contains requirements for elements of CIM_OpaqueManagementData. The requirements in
 1000 Table 36 are in addition to those specified for the superclass CIM_StorageExtent in Table 38.

1001 **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.

1002 **10.3 CIM_OpaqueManagementDataCapabilities**

1003 Table 37 contains requirements for elements of CIM_OpaqueManagementDataCapabilities. These
 1004 requirements are in addition to those specified for the superclass
 1005 CIM_EnabledLogicalElementCapabilities in the [Enabled Logical Element Profile](#).

1006 **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.

1007 **10.4 CIM_StorageExtent**

1008 Table 38 contains requirements for elements of CIM_StorageExtent. These requirements are in addition
 1009 to those specified for superclass CIM_EnabledLogicalElement in the [Enabled Logical Element Profile](#).

1010 **Table 38 – Class: CIM_StorageExtent**

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

1011 **10.5 CIM_PhysicalMedia**

1012 Table 39 contains the requirements for elements of CIM_PhysicalMedia.

1013 **Table 39 – Class: CIM_PhysicalMedia**

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

1014 **10.6 CIM_PhysicalMemory**

1015 An implementation that supports CIM_PhysicalMemory as specified in clause 7.3.11 shall support the
 1016 CIM_PhysicalMemory element requirements defined in the [Physical Asset Profile](#).

1017 **10.7 CIM_RegisteredProfile**

1018 An implementation shall support CIM_RegisteredProfile. With the exception of the mandatory values
 1019 specified for the properties in Table 40, the behavior of the CIM_RegisteredProfile instance is defined by
 1020 the [Profile Registration Profile](#).

1021 **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).

1022 **10.8 CIM_HostedService**

1023 Table 41 contains the requirements for elements of CIM_HostedService.

1024 **Table 41 – Class: CIM_HostedService**

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

1025 **10.9 CIM_ServiceAffectsElement (CIM_OpaqueManagementData)**

1026 Table 42 contains the element requirements for CIM_ServiceAffectsElement between
 1027 CIM_OpaqueManagementDataService and CIM_OpaqueManagementData.

1028 **Table 42 – Class: CIM_ServiceAffectsElement**

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

1029 **10.10 CIM_ServiceAffectsElement (CIM_StorageExtent)**

1030 Table 43 contains the element requirements for CIM_ServiceAffectsElement between
 1031 CIM_OpaqueManagementDataService and CIM_StorageExtent.

1032 **Table 43 – Class: CIM_ServiceAffectsElement**

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

1033 **10.11 CIM_ServiceAffectsElement (CIM_Identity)**

1034 Table 44 contains the element requirements for CIM_ServiceAffectsElement between
 1035 CIM_OpaqueManagementDataService and CIM_Identity.

1036 **Table 44 – Class: CIM_ServiceAffectsElement**

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

1037 **10.12 CIM_ServiceAffectsElementWithQuota**

1038 Table 45 contains the requirements for elements of CIM_ServiceAffectsElementWithQuota. The
 1039 requirements in Table 45 are in addition to those specified for the superclass CIM_ServiceAffectsElement
 1040 in Table 42.

1041 **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.

1042 **10.13 CIM_AssociatedPrivilege**

1043 Table 46 contains the requirements for elements of CIM_AssociatedPrivilege.

1044 **Table 46 – Class: CIM_AssociatedPrivilege**

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

1045 **10.14 CIM_OpaqueManagementDataOwner**

1046 No properties are required beyond those required for CIM_AssociatedPrivilege in 10.13.

1047 **10.15 CIM_BasedOn**

1048 Table 47 contains the requirements for elements of CIM_BasedOn.

1049 **Table 47 – Class: CIM_BasedOn**

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

1050 **10.16 CIM_ElementCapabilities**

1051 Table 48 contains the requirements for elements of CIM_ElementCapabilities.

1052 **Table 48 – Class: CIM_ElementCapabilities**

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

1053 **10.17 CIM_RealizesExtent**

1054 Table 49 contains the requirements for elements of CIM_RealizesExtent.

1055 **Table 49 – Class: CIM_RealizesExtent**

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

ANNEX A
(Informative)**Change Log**

Version	Date	Description
1.0.0	05-20-2009	DMTF Standard Release

1056
1057
1058
1059
1060

1061