



1
2
3
4
5
6

Document Number: DSP1081

Date: 2012-08-21

Version: 1.0.0

7 **Virtual System Migration Profile**

8 **Document Type: Specification**

9 **Document Status: DMTF Standard**

10 **Document Language: en-US**

11

12 Copyright Notice

13 Copyright © 2012 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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29 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

36	Foreword	6
37	Introduction.....	7
38	Document conventions.....	7
39	Typographical conventions	7
40	1 Scope	9
41	2 Normative references	9
42	3 Terms and definitions	10
43	4 Symbols and abbreviated terms.....	11
44	5 Synopsis	12
45	6 Description	13
46	6.1 Virtual system migration class schema.....	13
47	6.2 Virtual system migration concepts	15
48	6.2.1 Static migration	15
49	6.2.2 Dynamic migration	15
50	6.2.3 Live migration.....	16
51	7 Implementation.....	16
52	7.1 Virtual system migration service	16
53	7.2 Virtual system migration capabilities.....	16
54	7.2.1 General	17
55	7.2.2 Advertisement of method implementations	17
56	7.3 Method parameterization	17
57	7.3.1 CIM_VirtualSystemMigrationSettingData.MigrationType (Parameter)	17
58	7.3.2 CIM_VirtualSystemMigrationSettingData.Priority (Parameter).....	18
59	7.3.3 CIM_VirtualSystemMigrationSettingData.Bandwidth (Parameter)	18
60	7.3.4 CIM_VirtualSystemMigrationSettingData.BandwidthUnit (Parameter).....	18
61	7.3.5 CIM_VirtualSystemMigrationSettingData.TransportType (Parameter)	18
62	7.3.6 CIM_VirtualSystemMigrationSettingData.OtherTransportType (Parameter)	18
63	7.4 Migration settings representing capabilities.....	18
64	7.4.1 Default migration settings	18
65	7.4.2 Admissible migration settings	19
66	7.4.3 CIM_VirtualSystemMigrationSettingData.MigrationType (Capabilities)	20
67	7.4.4 CIM_VirtualSystemMigrationSettingData.Priority (Capabilities)	20
68	7.4.5 CIM_VirtualSystemMigrationSettingData.Bandwidth (Capabilities)	20
69	7.4.6 CIM_VirtualSystemMigrationSettingData.BandwidthUnit (Capabilities)	20
70	7.4.7 CIM_VirtualSystemMigrationSettingData.TransportType (Capabilities).....	20
71	7.4.8 CIM_VirtualSystemMigrationSettingData.OtherTransportType (Capabilities)	20
72	7.5 Virtual system correlation	21
73	7.5.1 General	21
74	7.5.2 CIM_ComputerSystem.OtherIdentifyingInfo[].....	21
75	7.5.3 CIM_ComputerSystem.IdentifyingDescriptions[]	21
76	7.5.4 Example	21
77	7.6 Indications.....	22
78	8 Methods.....	22
79	8.1 Extrinsic methods.....	22
80	8.1.1 CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost().....	22
81	8.1.2 CIM_VirtualSystemMigrationService.MigrateVirtualSystemToSystem()	23
82	8.1.3 CIM_VirtualSystemMigrationService.CheckVirtualSystemIsMigratableToHost().....	25
83	8.1.4 CIM_VirtualSystemMigrationService. CheckVirtualSystemIsMigratableToSystem()	26
84	8.2 Profile conventions for operations	27
85	8.2.1 CIM_AffectedJobElement	28
86	8.2.2 CIM_AssociatedJobMethodResult.....	28
87		

88	8.2.3	CIM_ConcreteJob	28
89	8.2.4	CIM_ComputerSystem	29
90	8.2.5	CIM_ElementCapabilities	29
91	8.2.6	CIM_HostedService	29
92	8.2.7	CIM_MethodResult	29
93	8.2.8	CIM_OwningJobElement	29
94	8.2.9	CIM_RegisteredProfile	30
95	8.2.10	CIM_ServiceAffectsElement	30
96	8.2.11	CIM_SettingsDefineCapabilities	30
97	8.2.12	CIM_VirtualSystemMigrationCapabilities	30
98	8.2.13	CIM_VirtualSystemMigrationService	30
99	8.2.14	CIM_VirtualSystemMigrationSettingData	30
100	9	Use-cases	30
101	9.1	Detection and inspection	31
102	9.1.1	Determine availability of migration services for a virtual system	31
103	9.1.2	Determine the capabilities of a virtual system migration service	31
104	9.1.3	Determine life migratability of a virtual system to a target virtualization platform	34
105	9.2	Migration operations	34
106	9.2.1	Live migration	34
107	9.2.2	Static migration with request for additional resources	37
108	10	CIM elements	38
109	10.1	CIM_AffectedJobElement	39
110	10.2	CIM_AssociatedJobMethodResult	40
111	10.3	CIM_ConcreteJob	40
112	10.4	CIM_ComputerSystem	41
113	10.5	CIM_ElementCapabilities	41
114	10.6	CIM_Error	42
115	10.7	CIM_HostedService	42
116	10.8	CIM_MethodResult	43
117	10.9	CIM_OwningJobElement	43
118	10.10	CIM_RegisteredProfile	44
119	10.11	CIM_ServiceAffectsElement	44
120	10.12	CIM_SettingsDefineCapabilities	45
121	10.13	CIM_VirtualSystemMigrationCapabilities	45
122	10.14	CIM_VirtualSystemMigrationService	45
123	10.15	CIM_VirtualSystemMigrationSettingData (Parameter)	46
124	10.16	CIM_VirtualSystemMigrationSettingData (Capabilities)	46
125	10.17	CIM_InstCreation	47
126	10.18	CIM_InstDeletion	47
127	10.19	CIM_InstMethodCall	48
128	10.20	CIM_InstModification	49
129	ANNEX A (informative)	Change log	51

130

131 Figures

132	Figure 1 – Virtual System Migration Profile: Profile class diagram	14
133	Figure 2 – Instance diagram: Virtual system migration capabilities	32
134	Figure 3 – Instance diagram: Virtual system migration	37
135		

136 Tables

137	Table 1 – Related profiles	13
-----	----------------------------------	----

138 Table 2 – CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() Method: Parameters 22

139 Table 3 – MigrateVirtualSystemToHost(): Standard messages..... 23

140 Table 4 – CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() Method: Parameters 24

141 Table 5 – MigrateVirtualSystemToSystem(): Standard messages 24

142 Table 6 – CIM_VirtualSystemMigrationService. CheckVirtualSystemIsMigratableToHost () Method:

143 Parameters..... 25

144 Table 7 – CheckVirtualSystemIsMigratableToHost(): Standard messages..... 26

145 Table 8 – CIM_VirtualSystemMigrationService.CheckVirtualSystemIsMigratableToSystem() Method:

146 Parameters..... 26

147 Table 9 – CheckVirtualSystemIsMigratableToSystem(): Standard messages 27

148 Table 10 – Operations: CIM_AffectedJobElement 28

149 Table 11 – Operations: CIM_AssociatedJobMethodResult 28

150 Table 12 – Operations: CIM_ElementCapabilities 29

151 Table 13 – Operations: CIM_HostedService 29

152 Table 14 – Operations: CIM_OwningJobElement..... 29

153 Table 15 – Operations: CIM_ServiceAffectsElement 30

154 Table 16 – Operations: CIM_SettingsDefineCapabilities..... 30

155 Table 17 – CIM Elements: Virtual System Migration profile 39

156 Table 18 – Association: CIM_AffectedJobElement..... 40

157 Table 19 – Association: CIM_AssociatedJobMethodResult 40

158 Table 20 – Class: CIM_ConcreteJob 41

159 Table 21 – Class: CIM_ComputerSystem..... 41

160 Table 22 – Association: CIM_ElementCapabilities 42

161 Table 23 – Class: CIM_Error..... 42

162 Table 24 – Association: CIM_HostedService..... 43

163 Table 25 – Class: CIM_MethodResult 43

164 Table 26 – Association: CIM_OwningJobElement..... 44

165 Table 27 – Class: CIM_RegisteredProfile 44

166 Table 28 – Association: CIM_ServiceAffectsElement..... 44

167 Table 29 – Association: CIM_SettingsDefineCapabilities 45

168 Table 30 – Class: CIM_VirtualSystemMigrationCapabilities..... 45

169 Table 31 – Class: CIM_VirtualSystemMigrationService 46

170 Table 32 – Class: CIM_VirtualSystemMigrationSettingData (Parameter) 46

171 Table 33 – Class: CIM_VirtualSystemMigrationSettingData (Capabilities) 47

172 Table 34 – Indication: CIM_InstCreation..... 47

173 Table 35 – Indication: CIM_InstDeletion 48

174 Table 36 – Indication: CIM_InstMethodCall 49

175 Table 37 – Indication: CIM_InstModification 49

176

177

Foreword

178 This profile - the *Virtual System Migration Profile* (DSP1081) - was prepared by the System Virtualization,
179 Partitioning and Clustering Working Group of the DMTF.

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

182 The DMTF acknowledges the following individuals for their contributions to this document:

183 • Editor:

184 – Michael Johanssen – IBM

185 • Participants from the DMTF System Virtualization, Partitioning and Clustering Working Group:

186 – Gareth Bestor – IBM

187 – Jim Fehlig – Novell

188 – Mark Hapner – Sun Microsystems, Inc.

189 – Ron Goering – IBM

190 – Steve Hand – Symantec Corporation

191 – Daniel Hiltgen – VMware Inc.

192 – Michael Johanssen – IBM

193 – Larry Lamers – VMware Inc.

194 – Andreas Maier - IBM

195 – Aaron Merkin – IBM

196 – John Parchem – Microsoft Corporation

197 – Shishir Pardikar – Citrix Systems Inc.

198 – Nihar Shah – Microsoft Corporation

199 – David Simpson – IBM

200

201

Introduction

202 The information in this specification should be sufficient for a provider or consumer of this data to identify
203 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
204 represent and manage the components described in this document. The target audience for this
205 specification is implementers who are writing CIM-based providers or consumers of management
206 interfaces that represent the components described in this document.

207 **Document conventions**

208 **Typographical conventions**

209 The following typographical conventions are used in this document:

- 210 • Document titles are marked in *italics*.
- 211 • ABNF rules are in `monospaced font`.

212

214

Virtual System Migration Profile

215 1 Scope

216 This profile is a component DMTF management profile that extends the management capabilities of the
217 referencing profile by adding the support to manage the migration of virtual systems. The support
218 includes functionality to initiate and control migration operations, and feasibility checks for a potential
219 migration operation.

220 2 Normative references

221 The following referenced documents are indispensable for the application of this document. For dated or
222 versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies.
223 For references without a date or version, the latest published edition of the referenced document
224 (including any corrigenda or DMTF update versions) applies.

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

227 DMTF DSP0200, *CIM Operations over HTTP 1.3*
228 http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

229 DMTF DSP0223, *Generic Operations 1.0*,
230 http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

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

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

235 DMTF DSP1042, *System Virtualization Profile 1.0*
236 http://www.dmtf.org/standards/published_documents/DSP1042_1.0.pdf

237 DMTF DSP1052, *Computer System Profile 1.0*
238 http://www.dmtf.org/standards/published_documents/DSP1052_1.0.pdf

239 DMTF DSP1054, *Indications Profile 1.1*
240 http://www.dmtf.org/standards/published_documents/DSP1054_1.1.pdf

241 DMTF DSP1057, *Virtual System Profile 1.0*
242 http://www.dmtf.org/standards/published_documents/DSP1057_1.0.pdf

243 DMTF DSP1103, *Job Control Profile 1.0*
244 http://www.dmtf.org/standards/published_documents/DSP1103_1.0.pdf

245 DMTF DSP8026, *System Virtualization Message Registry 1.0*
246 http://schemas.dmtf.org/wbem/messageregistry/1/dsp8026_1.0.xml

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

249 **3 Terms and definitions**

250 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
251 are defined in this clause.

252 The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"),
253 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
254 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
255 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
256 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
257 alternatives shall be interpreted in their normal English meaning.

258 The terms "clause", "subclause", "paragraph", and "annex" in this document are to be interpreted as
259 described in [ISO/IEC Directives, Part 2](#), Clause 5.

260 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC](#)
261 [Directives, Part 2](#), Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
262 not contain normative content. Notes and examples are always informative elements.

263 The terms defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. For the purposes of this
264 document, the terms and definitions given in [DMTF DSP1033](#) and [DMTF DSP1001](#) also apply. The
265 following additional terms are used in this document.

266 **3.1**

267 **conditional**

268 indicates requirements strictly to be followed in order to conform to the document and from which no
269 deviation is permitted when the specified conditions are met

270 **3.2**

271 **mandatory**

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

274 **3.3**

275 **optional**

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

277 **3.4**

278 **referencing profile**

279 indicates a profile that owns the definition of this class and can include a reference to this profile in its
280 "Related Profiles" table

281 **3.5**

282 **unspecified**

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

284 **3.6**

285 **implementation**

286 set of CIM providers that realize the classes specified by this profile

287 **3.7**

288 **client**

289 application that exploits facilities specified by this profile

- 290 **3.8**
291 **source virtualization platform**
292 the source of the virtual system being acted upon
- 293 **3.9**
294 **subject virtual system**
295 the virtual system that is being acted upon
- 296 **3.10**
297 **target virtualization platform**
298 the destination of the virtual system being acted upon
- 299 **3.11**
300 **this profile**
301 this DMTF management profile – the *Virtual System Migration Profile*
- 302 **3.12**
303 **virtual system migration**
304 process of moving a virtual system from a source virtualization platform to a target virtualization platform;
305 for details, see 6.2.
- 306 **3.13**
307 **virtual system migration service**
308 service that provides virtual system migration facilities as defined by this standard; for details, see 7.1.
- 309 **3.14**
310 **virtual computer system**
311 **virtual system**
312 concept of virtualization as applied to a computer system
- 313 Other common industry terms are *virtual machine*, *hosted computer*, *child partition*, *logical partition*,
314 *domain*, *guest*, or *container*.
- 315 **3.15**
316 **virtual system state**
317 state of a virtual system; for details, see ..
- 318 **3.16**
319 **virtualization platform**
320 virtualizing infrastructure provided by a host system enabling the provisioning and deployment of virtual
321 systems
- 322 **3.17**
323 **virtual system migration task**
324 task that performs a particular virtual system migration
- 325 **3.18**
326 **virtual system migration type**
327 type of virtual system migration; for details, see 6.2.

328 **4 Symbols and abbreviated terms**

329 The abbreviations defined in [DSP0004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following
330 additional abbreviations are used in this document.

- 331 **4.1**
- 332 **CIM**
- 333 Common Information Model
- 334 **4.2**
- 335 **CIMOM**
- 336 CIM object manager
- 337 **4.3**
- 338 **RASD**
- 339 CIM_ResourceAllocationSettingData
- 340 **4.4**
- 341 **SLP**
- 342 Service Location Protocol
- 343 **4.5**
- 344 **VS**
- 345 virtual system
- 346 **4.6**
- 347 **VSSD**
- 348 CIM_VirtualSystemSettingData
- 349 **4.7**
- 350 **VS_MIGRATION_METHOD_CALL**
- 351 the event that marks either the begin or the completion of a virtual system migration method call
- 352 **4.8**
- 353 **VS_MIGRATION_JOB_CREATE**
- 354 the event that marks a the creation of an instance of the CIM_ConcreteJob class representing a virtual
- 355 system migration task
- 356 **4.9**
- 357 **VS_MIGRATION_JOB_CHANGE**
- 358 the event that marks a change an instance of the CIM_ConcreteJob class representing a virtual system
- 359 migration task
- 360 **4.10**
- 361 **VS_MIGRATION_JOB_DELETE**
- 362 the event that marks the deletion of an instance of the CIM_ConcreteJob class representing a virtual
- 363 system migration task

364 **5 Synopsis**

365 **Profile Name:** *Virtual System Migration Profile*

366 **Version:** 1.0.0

367 **Organization:** DMTF

368 **CIM Schema Version:** 2.33

369 **Central Class:** CIM_VirtualSystemMigrationService

370 **Scoping Class:** CIM_System

371 This profile is a component profile that defines the minimum object model needed to provide for the
 372 migration of virtual systems.

373 Table 1 lists DMTF management profiles that this profile depends on.

374 **Table 1 – Related profiles**

Profile Name	Organization	Version	Relationship	Description
<i>Profile Registration</i>	DMTF	1.0	Mandatory	The profile that specifies registered profiles.
<i>Indications</i>	DMTF	1.0	Conditional ¹	The profile that specifies indications.
<i>Job Control</i>	DMTF	1.0	Optional	The profile that specifies job control.

¹ Condition: The implementation of [DMTF DSP1054](#) (*Indications Profile*) is required in the scope of the referencing profile if the indications defined in this profile are implemented.

375 **6 Description**

376 The *Virtual System Migration Profile* specifies the modeling of virtual system migration.

377 **6.1 Virtual system migration class schema**

378 Figure 1 shows the class schema of this profile. It outlines the elements that are owned or specialized by
 379 this profile, as well as the dependency relationships between elements of this profile and other profiles.
 380 For simplicity in diagrams the prefix CIM_ has been removed from class and association names.
 381

- 402 • the CIM_ElementCapabilities association modeling the relationship between an instance of the
403 CIM_VirtualSystemMigrationService class representing a virtual system migration service and
404 and instance of the the CIM_VirtualSystemMigrationCapabilities class representing its
405 capabilities
- 406 • the CIM_ConcreteJob class modeling a virtual system migration task, the
407 CIM_OwningJobElement association modeling the relationship between a virtual system
408 migration service and a virtual system migration task, and the CIM_AffectedJobElement
409 association modeling the relationship between a virtual system migration task and the affected
410 virtual system

411 In general, any mention of a class in this document means the class itself or its subclasses. For example,
412 a statement such as “an instance of the CIM_VirtualSystemMigrationSettingData class” implies an
413 instance of the CIM_VirtualSystemMigrationSettingData class or of a subclass of the
414 CIM_VirtualSystemMigrationSettingData class.

415 6.2 Virtual system migration concepts

416 Virtual system migration is the process of moving a virtual system from a source virtualization platform to
417 a target virtualization platform, such that after migration the virtual system can be used at the target
418 virtualization platform just as it was used at the source virtualization platform.

419 Three types of migration are distinguished:

- 420 • Static migration
- 421 • Dynamic migration
- 422 • Live migration

423 The implementation of a particular virtual system migration type depends on both the source and the
424 target virtualization platform. Consequently the set of implemented virtual system migration types is not a
425 static property of either source or target virtualization platform; instead the implementation of a particular
426 virtual system migration type needs to be determined dynamically by means of check methods that are
427 modeled as part of the virtual system migration service.

428 6.2.1 Static migration

429 Static migration applies primarily to the virtual system definition. Subject virtual systems are considered to
430 be in the in the “Defined” virtual system state, that is, they are not instantiated and not performing work.

431 Static migration comprises activities such as

- 432 • validating that source and target virtualization platform are compatible
- 433 • validating that external connections, such as access to network and storage resources defined
434 for the virtual system at the source virtualization platform can be identically redefined at the
435 target virtualization platform
- 436 • redefining the virtual system at the target virtualization platform, including definition of external
437 connections
- 438 • moving persistent resources like the content of locally defined virtual disks from the source
439 virtualization platform to the target virtualization platform
- 440 • removing the virtual system from the source virtualization platform

441 6.2.2 Dynamic migration

442 Dynamic migration applies to both the virtual system definition and to the virtual system instance. Subject
443 virtual systems may be in the “Defined”, “Active”, “Paused” or “Suspended” virtual system state. The
444 “Active” virtual system state implies that the virtual system is instantiated and actively performs tasks; but

445 while the virtual system may be active when a dynamic migration process is initiated, the virtual system
446 (including all of its virtual resources) is prevented from performing any activities for the complete duration
447 of the migration process.

448 Dynamic migration comprises of the activities required for static migration, but in addition requires
449 activities affecting the virtual system instance, like

- 450 • re-instantiating the virtual system at the target virtualization platform
- 451 • establishing external connections that were in effect for the virtual system at the source
452 virtualization platform identically at the target virtualization platform
- 453 • moving volatile resources like the content of virtual memory from the source virtualization
454 platform to the target virtualization platform

455 **6.2.3 Live migration**

456 Live migration applies to virtual system definition and virtual system instance. Subject virtual systems may
457 be in the “Defined”, “Active”, “Paused” or “Suspended” virtual system states, that is, they may be
458 instantiated and may continue performing work while the migration process is in progress.

459 Live migration comprises of activities required for static and for dynamic migration. It requires closer
460 coordination between source and target virtualization platform such that state changes resulting from
461 activities performed by the virtual system while the migration is in progress are captured. The integrity of
462 the virtual system, its components, and its external connections is maintained over the whole migration
463 process. External clients of software executed within the virtual system do not notice the ongoing
464 migration process aside from performance impacts.

465 **7 Implementation**

466 This clause details the requirements related to classes and their properties for implementations of this
467 profile. The CIM Schema descriptions for any referenced element and its sub-elements apply.

468 The list of all methods covered by this profile is in clause 8. The list of all properties covered by this profile
469 is in clause 10.

470 In references to CIM Schema properties that enumerate values, the numeric value is normative and the
471 descriptive text following it in parenthesis is informational. For example, in the statement “If an instance of
472 the CIM_VirtualSystemMigrationCapabilities class contains the value 3 (MigrateVirtualSystemToSystem)
473 in an element of the SynchronousMethodsSupported[] array property”, the “value 3” is normative text and
474 “(MigrateVirtualSystemToSystem)” is descriptive text.

475 **7.1 Virtual system migration service**

476 The CIM_VirtualSystemMigrationService class is used to model virtual system migration services. Each
477 virtual system migration service shall be represented by one instance of the
478 CIM_VirtualSystemMigrationService class. A virtual system migration service is required to provide
479 functionality for synchronous or asynchronous virtual system migration or both, such that this functionality
480 can be exposed by the methods of the CIM_VirtualSystemMigrationService class.

481 **7.2 Virtual system migration capabilities**

482 This subclause details the requirements related to representing the capabilities of a virtual system
483 migration service.

484 7.2.1 General

485 The CIM_VirtualSystemMigrationCapabilities class is used to model the capabilities of virtual migration
 486 services. The capabilities of a virtual system migration service shall be represented by an instance of the
 487 CIM_VirtualSystemMigrationCapabilities class. That instance shall be associated to the instance of the
 488 CIM_VirtualSystemMigrationService class representing the virtual system migration service (see 7.1)
 489 through an instance of the CIM_ElementCapabilities association.

490 7.2.2 Advertisement of method implementations

491 Values defined in the ValueMap qualifier of the SynchronousMethodsSupported[] and
 492 AsynchronousMethodsSupported[] array properties of the CIM_VirtualSystemMigrationCapabilities class
 493 shall designate corresponding methods of the CIM_VirtualSystemMigrationService class, as follows:

- 494 • The value 2 (MigrateVirtualSystemToHostSupported) shall designate the
 495 MigrateVirtualSystemToHost() method
- 496 • The value 3 (MigrateVirtualSystemToSystemSupported) shall designate
 497 MigrateVirtualSystemToSystem() method
- 498 • The value 4 (CheckVirtualSystemIsMigratableToHostSupported) shall designate the
 499 CheckVirtualSystemIsMigratableToHost () method
- 500 • The value 5 (CheckVirtualSystemIsMigratableToSystemSupported) shall designate the
 501 CheckVirtualSystemIsMigratableToSystem() method

502 The following rules apply to the use of these values for elements of the SynchronousMethodsSupported[]
 503 or AsynchronousMethodsSupported[] array properties in an instance of the
 504 CIM_VirtualSystemMigrationCapabilities class that represents the capabilities of a virtual system
 505 migration service:

- 506 • If a particular method is not implemented, the corresponding qualifier value shall not be present
 507 as a value for an element in either array
- 508 • If a particular method is implemented with synchronous behavior, exactly one element of the
 509 SynchronousMethodsSupported[] array property shall have the value that corresponds to that
 510 method as defined in this subclause
- 511 • If a particular method is implemented with asynchronous behavior, exactly one element of the
 512 AsynchronousMethodsSupported[] array property shall have the value that corresponds to that
 513 method as defined in this subclause

514 NOTE The normative text requires that methods implementing both synchronous and asynchronous behavior are
 515 indicated through both array properties.

516 NOTE A client discovers the fact that asynchronous execution of methods is implemented for one or more methods
 517 by checking the value of the AsynchronousMethodsSupported[] array property in the instance of the
 518 CIM_VirtualSystemMigrationCapabilities class representing the capabilities of a virtual system migration
 519 service is not NULL, and contains at least one non-NULL element.

520 7.3 Method parameterization

521 Instances of the CIM_VirtualSystemMigrationSettingData class shall be supported as a value of the
 522 MigrationSettingData parameter of methods of the CIM_VirtualSystemMigrationService class. If no
 523 instance or an incomplete instance is provided by the client, default values shall be applied as specified in
 524 7.4.

525 7.3.1 CIM_VirtualSystemMigrationSettingData.MigrationType (Parameter)

526 If specified, the value of the MigrationType property shall be used to determine the requested migration
 527 type. If not specified, the respective default value shall be used to determine the migration type; see
 528 7.4.3.

529 **7.3.2 CIM_VirtualSystemMigrationSettingData.Priority (Parameter)**

530 If specified, the value of the Priority property shall be used to determine the requested migration priority. If
531 not specified, the respective default value shall be used to determine the migration priority; see 7.4.4.

532 **7.3.3 CIM_VirtualSystemMigrationSettingData.Bandwidth (Parameter)**

533 If specified, the value of the Bandwidth property shall be used to determine the requested bandwidth for
534 the migration process. If not specified, the respective default value shall be used to determine the
535 bandwidth; see 7.4.5.

536 **7.3.4 CIM_VirtualSystemMigrationSettingData.BandwidthUnit (Parameter)**

537 If specified, the value of the BandwidthUnit property shall be used to determine the unit of the requested
538 bandwidth for the migration process. If not specified, the respective default value shall be used to
539 determine the bandwidth unit; see 7.4.6.

540 **7.3.5 CIM_VirtualSystemMigrationSettingData.TransportType (Parameter)**

541 If specified, the value of the TransportType property shall be used to determine the requested transport
542 type for the migration process. If not specified, the respective default value shall be used to determine the
543 transport type; see 7.4.7.

544 **7.3.6 CIM_VirtualSystemMigrationSettingData.OtherTransportType (Parameter)**

545 The implementation of the OtherTransportType property is conditional.

546 Condition: The TransportType property is implemented, and the value of 1 (Other) is supported.

547 If the value of the TransportType property is 1 (Other), and the value of the OtherTransportType is
548 specified, the value of the OtherTransportType property shall be used to determine the requested
549 transport type for the migration process. If not specified, the respective default value shall be used to
550 determine the transport type; see 7.4.6 and 7.4.8.

551 **7.4 Migration settings representing capabilities**

552 This subclause specifies the use of the CIM_VirtualSystemMigrationSettingData class and the
553 CIM_SettingsDefineCapabilities association for the representation of default migration settings, and for
554 the representation of valid migration settings that are accepted by methods of the related virtual system
555 migration service.

556 **7.4.1 Default migration settings**

557 An instance of the CIM_VirtualSystemMigrationCapabilities class representing capabilities of a virtual
558 system migration service shall be associated to an instance of the
559 CIM_VirtualSystemMigrationSettingData class through an instance of the CIM_SettingsDefineCapabilities
560 association.

561 Properties in the instance of the CIM_SettingsDefineCapabilities association shall be set as follows:

- 562 • The value of the PropertyPolicy property shall be 0 (Independent)
- 563 • The value of the ValueRole property shall be 0 (Default)
- 564 • The value of the ValueRange property shall be 0 (Point)

565 The instance of the CIM_VirtualSystemMigrationSettingData class shall convey information about the
566 default migration settings that apply if one of the migration methods of the virtual system migration service
567 is called and no value is provided for the MigrationSettingData parameter or if an instance of the

568 CIM_VirtualSystemMigrationSettingData class is provided as value of the MigrationSettingData
569 parameter, but not all properties were assigned values.

570 **7.4.2 Admissible migration settings**

571 An implementation may provide instances of the CIM_VirtualSystemMigrationSettingData class that
572 describe admissible values or ranges of admissible values for methods of a virtual system migration
573 service.

574 If such instances are provided by an implementation, they shall be associated to the instance of the
575 CIM_VirtualSystemMigrationCapabilities class that describes the capabilities of the respective virtual
576 system migration service (see 7.2) through instances of the CIM_SettingsDefineCapabilities association
577 where properties shall be set as follows:

- 578 • The value of the PropertyPolicy property
 - 579 – shall match 0 (Independent) if the properties of the referenced instance of the
580 CIM_VirtualSystemMigrationSettingData class apply without correlation, that is, each
581 property value applies regardless of other property values in the referenced instance
 - 582 – shall match 1 (Correlated) if the properties of the referenced instance of the
583 CIM_VirtualSystemMigrationSettingData class apply with correlation, that is, each value
584 applies only in correlation to other property values in the referenced instance
- 585 • The value of the ValueRole property shall match 3 (Supported)
- 586 • The value of the ValueRange property shall match one of the following:
 - 587 – 0 (Point) if the referenced instance of the CIM_VirtualSystemMigrationSettingData class
588 represents one particular admissible setting. A particular instance of the
589 CIM_VirtualSystemMigrationCapabilities class may have zero or more instances of the
590 CIM_VirtualSystemMigrationSettingData class associated this way, expressing a set of
591 admissible values. If such instances are provided, methods of the related virtual system
592 migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData
593 class that match these instances.
 - 594 – 1 (Minimums) if the numeric values in the referenced instance of the
595 CIM_VirtualSystemMigrationSettingData class represent minimum values; in this case,
596 NULL values and values of non-numeric properties shall be ignored. A particular instance
597 of the CIM_VirtualSystemMigrationCapabilities class may have at most one instance of the
598 CIM_VirtualSystemMigrationSettingData class associated this way, expressing the
599 admissible minimum. If such instance is provided, methods of the related virtual system
600 migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData
601 class where the numeric values are above the minimum values as expressed by the
602 referenced instance.
 - 603 – 2 (Maximums) if the numeric values in the referenced instance of the
604 CIM_VirtualSystemMigrationSettingData class represent maximum values; in this case,
605 NULL values and values of non-numeric properties shall be ignored. A particular instance
606 of the CIM_VirtualSystemMigrationCapabilities class may have at most one instance of the
607 CIM_VirtualSystemMigrationSettingData class associated this way, expressing the
608 admissible maximum. If such instance is provided, methods of the related virtual system
609 migration service shall accept instances of the CIM_VirtualSystemMigrationSettingData
610 class where the numeric values are below the maximum values as expressed by the
611 referenced instance.
 - 612 – 3 (Increments) if the numeric values in the referenced instance of the
613 CIM_VirtualSystemMigrationSettingData class represent an increment; in this case, NULL
614 values and values of non-numeric properties shall be ignored. A particular instance of the
615 CIM_VirtualSystemMigrationCapabilities class may have at most one instance of the
616 CIM_VirtualSystemMigrationSettingData class associated this way, expressing the

617 admissible increment. If such instance is provided, a minimum and a maximum shall be
618 provided as specified in this subclause, and the increment shall apply between the
619 minimum and the maximum. If such instance is provided, methods of the related virtual
620 system migration service shall accept instances of the
621 CIM_VirtualSystemMigrationSettingData class where the numeric values are within the
622 range specified by the minimum and the maximum, and within that range match multiples
623 of the increment.

624 **7.4.3 CIM_VirtualSystemMigrationSettingData.MigrationType (Capabilities)**

625 The value of the MigrationType property shall convey the default migration type or an admissible
626 migration type applicable to migration processes initiated or checked through methods of a virtual system
627 migration service.

628 **7.4.4 CIM_VirtualSystemMigrationSettingData.Priority (Capabilities)**

629 The value of the Priority property shall be used to convey the default priority or an admissible priority
630 applicable to migration processes initiated or checked through methods of a virtual system migration
631 service.

632 **7.4.5 CIM_VirtualSystemMigrationSettingData.Bandwidth (Capabilities)**

633 The implementation of the Bandwidth property for the representation of the default bandwidth or an
634 admissible bandwidth is optional.

635 If implemented, the value of the Bandwidth property shall convey the default bandwidth or an admissible
636 bandwidth applicable to migration processes initiated or checked through methods of a virtual system
637 migration service.

638 **7.4.6 CIM_VirtualSystemMigrationSettingData.BandwidthUnit (Capabilities)**

639 The implementation of the Bandwidth property for the representation of the default bandwidth or an
640 admissible bandwidth unit is conditional.

641 Condition: The BandwidthUnit property shall be implemented if the Bandwidth property for the
642 representation of the default bandwidth or of an admissible bandwidth is implemented; see 7.4.5.

643 If implemented, the value of the BandwidthUnit property shall convey the default bandwidth unit
644 applicable to the value of the default bandwidth and for the bandwidth property in migration request if the
645 request does not specify a bandwidth unit.

646 **7.4.7 CIM_VirtualSystemMigrationSettingData.TransportType (Capabilities)**

647 The implementation of the TransportType property for the representation of the default transport type or
648 an admissible transport type is optional.

649 If implemented, the value of the TransportType property shall convey the default transport type applicable
650 to migration processes initiated or checked through methods of a virtual system migration service.

651 **7.4.8 CIM_VirtualSystemMigrationSettingData.OtherTransportType (Capabilities)**

652 The implementation of the OtherTransportType property for the representation of the default transport
653 type or an admissible transport type is conditional.

654 Condition: The TransportType property is implemented, and the value 1 (Other) is supported.

655 If the value of the TransportType property is 1 (Other), the value of the OtherTransportType property shall
656 be used to convey the default transport type applicable to migration processes initiated or checked
657 through methods of a virtual system migration service.

658 7.5 Virtual system correlation

659 This subclause details requirement with respect to the correlation of the instances of the
660 CIM_ComputerSystem class representing the source and the migrated virtual system.

661 7.5.1 General

662 The implementation of virtual system correlation is optional.

663 The new virtual system that is implicitly created as part of a virtual system migration process shall be
664 logically identical to the original virtual system. The logical identity relationship shall be represented
665 through values of the OtherIdentifyingInfo[] and IdentifyingDescriptions[] array properties such that at
666 least one value pair exposed by these properties in the instance of the CIM_ComputerSystem class
667 representing of the source virtual system is present in the instance of the CIM_ComputerSystem class
668 representing the target virtual system.

669 7.5.2 CIM_ComputerSystem.OtherIdentifyingInfo[]

670 Condition: The CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() method is
671 implemented.

672 The value of the OtherIdentifyingInfo[] array property in the instance of the CIM_ComputerSystem
673 representing the implicitly created migrated virtual system shall contain at least one element from the
674 value of the OtherIdentifyingInfo[] array property in the instance of the CIM_ComputerSystem class
675 representing the source virtual system.

676 7.5.3 CIM_ComputerSystem.IdentifyingDescriptions[]

677 Condition: If CIM_ComputerSystem.OtherIdentifyingInfo is implement then
678 CIM_ComputerSystem.IdentifyingDescription shall be implemented.

679 The values of elements in the value of the IdentifyingDescriptions[] array property that have the same
680 array index as those elements of the value of the OtherIdentifyingInfo[] array property that match the
681 specification in 7.5.2 shall be identical in the instance of the CIM_ComputerSystem representing the
682 implicitly created migrated virtual system and the instance of the CIM_ComputerSystem class
683 representing the source virtual system.

684 7.5.4 Example

685 For example, if in the instance of the CIM_ComputerSystem class representing a source virtual system

- 686 • the value of the OtherIdentifyingInfo[] array property is { "CIM:GUID", "CIM:MAC", "CIM:Tag" }
- 687 • the value of the IdentifyingDescriptions array property is { "01234...", "0123456789AB",
688 "CustomTag" }

689 then these values compose three value pairs that each provides a unique correlatable identification of the
690 source virtual system. The requirements specified in 7.5.2 and 7.5.3 require that in the instance of the
691 CIM_ComputerSystem class representing the migrated virtual system, the values of the
692 OtherIdentifyingInfo[] array property and the IdentifyingDescriptions[] array property exhibit at least one
693 of the value pairs defined in the instance of the CIM_ComputerSystem class representing the source
694 virtual system. For example, if in the instance of the CIM_ComputerSystem class representing the
695 migrated virtual system

- 696 • the value of the OtherIdentifyingInfo[] array property is { "CIM:Tag" }
- 697 • the value of the IdentifyingDescriptions array property is { "CustomTag" }

698 then the requirement would be complied with.

699 7.6 Indications

700 The implementation of indications is optional.

701 [DMTF DSP1054](#) (*Indications Profile*) defines indications as the means to communicate events. The
702 handling of events related to virtual system migration are defined in [DMTF DSP1103](#).

703 8 Methods

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

706 The CIM Schema descriptions for any referenced method and its parameters apply.

707 8.1 Extrinsic methods

708 This subclause details specifications for extrinsic methods of the `CIM_VirtualSystemMigrationService`
709 class.

710 8.1.1 `CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost()`

711 The implementation of the `MigrateVirtualSystemToHost()` method is optional; if implemented, the
712 requirements defined in 7.2.2 apply.

713 Condition: The implementation of the `MigrateVirtualSystemToHost()` method is required if the
714 `MigrateVirtualSystemToSystem()` method is not implemented; see 8.1.2.

715 Table 2 specifies detailed requirements for the `MigrateVirtualSystemToHost()` method.

716 **Table 2 – `CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost()` Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the <code>CIM_ComputerSystem</code> class representing the virtual system to be migrated.
IN	DestinationHost	string	Address of the destination host; for the format see CIM Schema description.
IN	MigrationSettingData	string	Embedded instance of the <code>CIM_VirtualSystemMigrationSettingData</code> class.
IN	NewSystemSettingData	string	Embedded instance of the <code>CIM_VirtualSystemSettingData</code> class that replaces or adds property values for the virtual system after it is migrated.
IN	NewResourceSettingData	string[]	Embedded instances of the <code>CIM_ResourceAllocationSettingData</code> class that replace or add property values for virtual resources.
OUT	Job	CIM_ConcreteJob REF	A reference to the job that performs the task (NULL if the task is completed on return).

717 The method shall either return a return code or an exception. For return code values, see the CIM
718 schema description.

719 The implementation of standard messages is optional. Table 3 specifies the optional standard messages
 720 for the MigrateVirtualSystemToHost() method.

721 **Table 3 – MigrateVirtualSystemToHost(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_EN> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC0101	The file <File_Name_Name> <Unexpected_File_State>.
(1) DMTF SVPC0102	The file <File_Name_Name> <Failed_File_Operation>.
(4096) DMTF SVPC8001	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has been initiated.
(1) DMTF SVPC8003	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has failed.
(0) DMTF SVPC8004	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has successfully completed.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.
(1) DMTF SVPC8040	The <Migration_Operation>() parameter <Parameter_Name> at index <Parameter_Index> is <Parameter_Error>; expected parameter value is <Parameter_Spec>.
(1) DMTF SVPC8041	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a <Component_Error> <Component_RC> occurred.
(1) DMTF SVPC8042	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a timeout occurred.

722 **8.1.2 CIM_VirtualSystemMigrationService.MigrateVirtualSystemToSystem()**

723 The implementation of the MigrateVirtualSystemToSystem() method is optional; if implemented, the
 724 requirements defined in 7.2.2 apply.

725 Table 4 specifies detailed requirements for the MigrateVirtualSystemToSystem() method.

726 **Table 4 – CIM_VirtualSystemMigrationService.MigrateVirtualSystemToHost() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.
IN	DestinationSystem	CIM_System REF	Reference to an instance of the CIM_System class representing the destination host.
IN	MigrationSettingData	String	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	String	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.
IN	NewResourceSettingData	string[]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	NewComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system after it has been migrated.
OUT	Job	CIM_ConcreteJob REF	A reference to the job that performs the task (NULL if the task is completed on return).

727 The method shall either return a return code or an exception. For return code values, see the CIM
728 schema description.

729 The implementation of standard messages is optional. Table 5 specifies the optional standard messages
730 for the MigrateVirtualSystemToSystem() method.

731 **Table 5 – MigrateVirtualSystemToSystem(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_Name> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC0101	The file <File_Name_Name> <Unexpected_File_State>.
(1) DMTF SVPC0102	The file <File_Name_Name> <Failed_File_Operation>.
(4096) DMTF SVPC8001	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has been initiated.
(1) DMTF SVPC8003	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to

(Return Code) MessageID	Message Text
	the host system named <Target_Host_Name> has failed.
(0) DMTF SVPC8004	The migration process migrating the virtual system named <Virtual_System_Name> from the host system named <Source_Host_Name> to the host system named <Target_Host_Name> has successfully completed.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.
(1) DMTF SVPC8040	The <Migration_Operation>() parameter <Parameter_Name> at index <Parameter_Index> is <Parameter_Error>; expected parameter value is <Parameter_Spec>.
(1) DMTF SVPC8041	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a <Component_Error> <Component_RC> occurred.
(1) DMTF SVPC8042	The migration of the virtual system named <Virtual_System_Name> to the target host system named <Target_Host_Name> failed because a timeout occurred.

732 **8.1.3 CIM_VirtualSystemMigrationService.CheckVirtualSystemsMigratableToHost()**

733 The implementation of the CheckVirtualSystemsMigratableToHost() method is conditional; if
734 implemented, the requirements defined in 7.2.2 apply.

735 Condition: The implementation of the CheckVirtualSystemsMigratableToHost() method is required if the
736 CheckVirtualSystemsMigratableToSystem() method is not implemented; see 8.1.4.

737 The method shall be implemented with synchronous behavior only.

738 Table 6 specifies detailed requirements for the CheckVirtualSystemsMigratableToHost() method.

739 **Table 6 – CIM_VirtualSystemMigrationService. CheckVirtualSystemsMigratableToHost () Method:**
740 **Parameters**

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.
IN	DestinationHost	string	Address of the destination host; for the format see CIM Schema description.
IN	MigrationSettingData	string	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	string	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.

Qualifiers	Name	Type	Description/Values
IN	NewResourceSettingData	string[]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	IsMigratable	boolean	Method result indicating whether the source virtual system is migratable to the target host.

741 The method shall either return a return code or an exception. For return code values, see the CIM
742 schema description.

743 The implementation of standard messages is optional. Table 7 specifies the optional standard messages
744 for the CheckVirtualSystemsIsMigratableToHost() method.

745 **Table 7 – CheckVirtualSystemsIsMigratableToHost(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_Name> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.

746 **8.1.4 CIM_VirtualSystemMigrationService.CheckVirtualSystemsMigratableToSystem()**

747 The implementation of the CheckVirtualSystemsMigratableToSystem() method is optional; if
748 implemented, the requirements defined in 7.2.2 apply.

749 The method shall be implemented with synchronous behavior only.

750 Table 8 specifies detailed requirements for the CheckVirtualSystemsMigratableToSystem() method.

751 **Table 8 – CIM_VirtualSystemMigrationService.CheckVirtualSystemsMigratableToSystem()**
752 **Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	ComputerSystem	CIM_ComputerSystem REF	Reference to an instance of the CIM_ComputerSystem class representing the virtual system to be migrated.

Qualifiers	Name	Type	Description/Values
IN	DestinationSystem	CIM_System REF	Reference to an instance of the CIM_System class representing the destination host.
IN	MigrationSettingData	string	Embedded instance of the CIM_VirtualSystemMigrationSettingData class.
IN	NewSystemSettingData	string	Embedded instance of the CIM_VirtualSystemSettingData class that replaces or adds property values for the virtual system after it is migrated.
IN	NewResourceSettingData	string[]	Embedded instances of the CIM_ResourceAllocationSettingData class that replace or add property values for virtual resources.
OUT	IsMigratable	boolean	Method result indicating whether the source virtual system is migratable to the target host.

753 The method shall either return a return code or an exception. For return code values, see the CIM
 754 schema description.

755 Support of standard messages is optional. Table 9 specifies the optional standard messages for the
 756 CheckVirtualSystemIsMigratableToSystem() method.

757 **Table 9 – CheckVirtualSystemIsMigratableToSystem(): Standard messages**

(Return Code) MessageID	Message Text
(6) DMTF SVPC0003	The virtual system named <Virtual_System_Name> does not exist.
(1) DMTF SVPC0004	The virtual system named <Virtual_System_Name> is in the <Virtual_System_State> state, but the requested operation requires one of the following virtual system states: <Required_Virtual_System_States>.
(1) DMTF SVPC0005	The virtual system named <Virtual_System_Name> has the virtual system type <Virtual_System_Type> state, but the requested operation requires one of the following virtual system types: <Required_Virtual_System_Types>.
(1) DMTF SVPC0006	The virtual system named <Virtual_System_Name> <Virtual_System_Config_Error> that prevent(s) the requested operation.
(1) DMTF SVPC8020	The target host named <Target_Host_Name><Target_Host_State>.
(1) DMTF SVPC8021	The target virtual system named <Virtual_System_Name> already exists.
(1) DMTF SVPC8022	The target virtual system named <Virtual_System_Name> <Failed_Activity>.
(1) DMTF SVPC8023	The target virtual system named <Virtual_System_Name> <Configuration_Error>.
(1) DMTF SVPC8030	The resource requirements of the virtual system named <Virtual_System_Name> for the <Resource_Type> cannot be satisfied at the target host system named <Target_Host_Name>.

758 **8.2 Profile conventions for operations**

759 Support for operations for each profile class (including associations) is specified in the following
 760 subclauses. Each subclause includes either a statement “All operations in the default list in subclause 8.2
 761 are supported as described by [DMTF DSP0200](#) or a table listing all of the operations that are not

762 supported by this profile or where the profile requires behavior other than that described by [DMTF](#)
 763 [DSP0200](#).

764 The default list of operations is as follows:

- 765 • GetInstance
- 766 • Associators
- 767 • AssociatorNames
- 768 • References
- 769 • ReferenceNames
- 770 • EnumerateInstances
- 771 • EnumerateInstanceNames

772 A compliant implementation shall support all of the operations in the default list for each class, unless the
 773 “Requirement” column states something other than *Mandatory*.

774 This profile defines methods in terms of [DMTF DSP0200](#).

775 8.2.1 CIM_AffectedJobElement

776 Table 10 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
 777 shall not be implemented.

778 **Table 10 – Operations: CIM_AffectedJobElement**

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

779 8.2.2 CIM_AssociatedJobMethodResult

780 Table 11 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
 781 shall not be implemented.

782 **Table 11 – Operations: CIM_AssociatedJobMethodResult**

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

783 8.2.3 CIM_ConcreteJob

784 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).

785 **8.2.4 CIM_ComputerSystem**

786 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).

787 **8.2.5 CIM_ElementCapabilities**

788 Table 12 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
789 shall not be implemented.

790 **Table 12 – Operations: CIM_ElementCapabilities**

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

791 **8.2.6 CIM_HostedService**

792 Table 13 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
793 shall not be implemented.

794 **Table 13 – Operations: CIM_HostedService**

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

795 **8.2.7 CIM_MethodResult**

796 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).

797 **8.2.8 CIM_OwningJobElement**

798 Table 14 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
799 shall not be implemented.

800 **Table 14 – Operations: CIM_OwningJobElement**

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

801 **8.2.9 CIM_RegisteredProfile**802 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).803 **8.2.10 CIM_ServiceAffectsElement**804 Table 15 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
805 shall not be implemented.806 **Table 15 – Operations: CIM_ServiceAffectsElement**

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

807 **8.2.11 CIM_SettingsDefineCapabilities**808 Table 16 lists operations that either have special requirements beyond those from [DMTF DSP0200](#) or
809 shall not be implemented.810 **Table 16 – Operations: CIM_SettingsDefineCapabilities**

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

811 **8.2.12 CIM_VirtualSystemMigrationCapabilities**812 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).813 **8.2.13 CIM_VirtualSystemMigrationService**814 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).815 **8.2.14 CIM_VirtualSystemMigrationSettingData**816 All operations in the default list in subclause 8.2 are supported as described by [DMTF DSP0200](#).817 **9 Use-cases**818 The following use-cases and object diagrams illustrate use of this profile. They are for informational
819 purposes only and do not introduce behavioral requirements for implementations of the profile.

820 9.1 Detection and inspection

821 This set of use-cases describes how to determine whether a conformant migration service is available for
822 a particular virtual system, and whether the virtual system is migratable to a particular target host system.

823 NOTE Use-cases describing the discovery of profile implementations and the detection of instances of the central
824 class are described in [DMTF DSP1033](#).

825 9.1.1 Determine availability of migration services for a virtual system

826 **Assumption:** The client knows a reference to an instance of the CIM_ComputerSystem class that
827 represents a virtual system.

828 • The client resolves the CIM_ServiceAffectsElement association to find the instance of the
829 CIM_VirtualSystemMigrationService class that represents the virtual system migration service
830 that is responsible for the virtual system, invoking the intrinsic AssociatorNames() CIM operation
831 with parameter values set as follows:

832 – the value of the ObjectName parameter refers to the instance of the CIM_ComputerSystem
833 class that represents the virtual system

834 – the value of the AssocClass parameter is set to “CIM_ServiceAffectsElement”

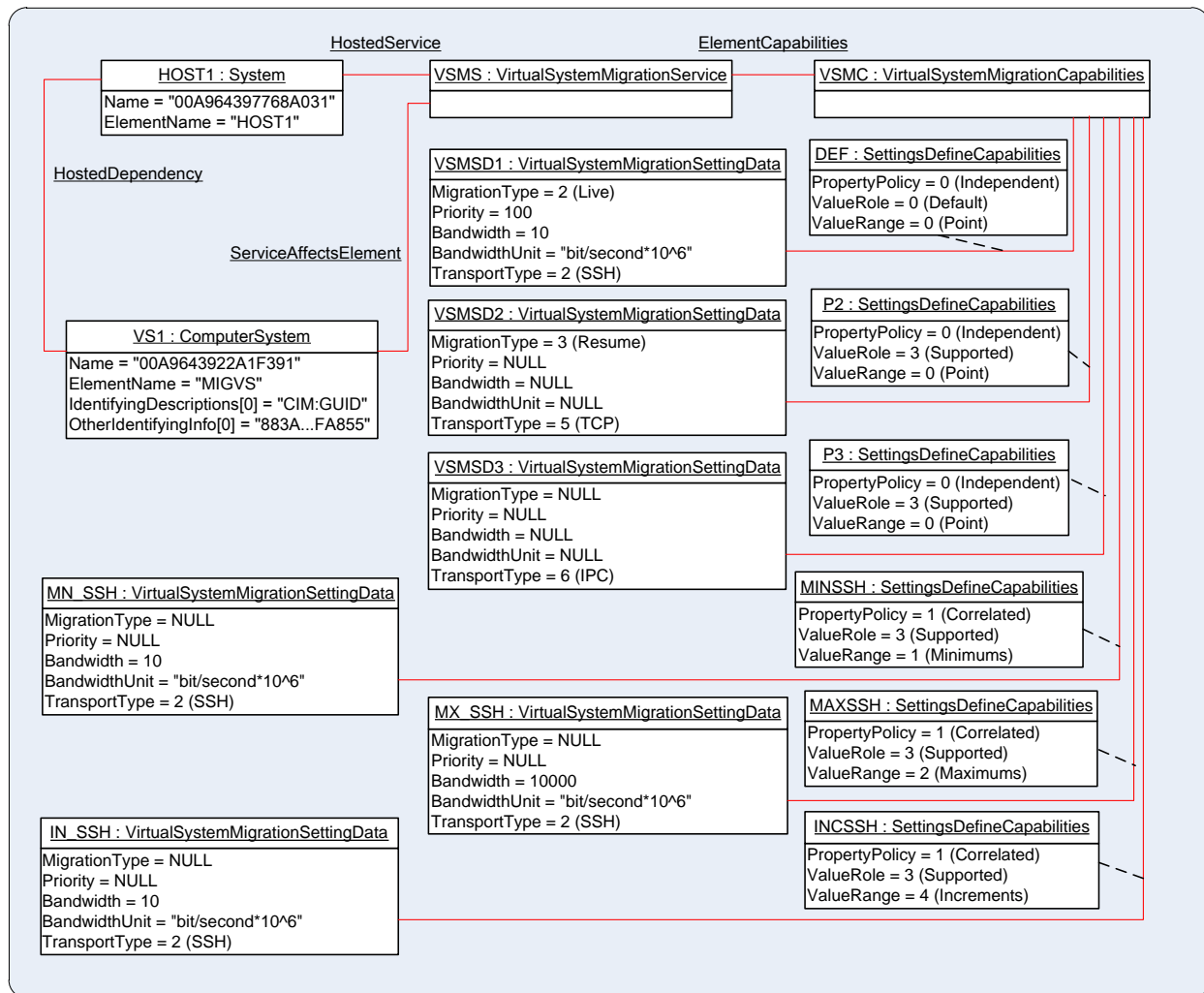
835 – the value of the ResultClass parameter is set to “CIM_VirtualSystemMigrationService”

836 The result is a set of references containing one element referencing the
837 CIM_VirtualSystemMigrationService class representing virtual system migration service.

838 **Result:** The client knows virtual system migration services that are responsible for the virtual system; see
839 [DMTF DSP1033](#) for recipes that describe the detection of profile conformant elements, i.e. a virtual
840 system migration service.

841 9.1.2 Determine the capabilities of a virtual system migration service

842 This use-case describes how to determine the capabilities of a virtual system migration service.



843

844

Figure 2 – Instance diagram: Virtual system migration capabilities

845 **Assumption:** The client knows a reference to an instance of the CIM_VirtualSystemMigrationService
 846 class representing a virtual system migration service.

847 1) The client resolves the CIM_ElementCapabilities association to find the instance of the
 848 CIM_VirtualSystemMigrationCapabilities class that represents the capabilities of the virtual
 849 system migration service, invoking the intrinsic AssociatorNames() CIM operation with
 850 parameter values set as follows:

- 851 – the value of the ObjectName parameter refers to the instance of the
- 852 CIM_VirtualSystemMigrationService class that represents the virtual system migration
- 853 service. In Figure 2, that input instance is the instance VSMS.
- 854 – the value of the AssocClass parameter is set to "CIM_ElementCapabilities"
- 855 – the value of the ResultClass parameter is set to "CIM_VirtualSystemMigrationCapabilities"

856 The result is a set of references containing one element referencing the
 857 CIM_VirtualSystemMigrationCapabilities class representing the capabilities of the virtual system
 858 migration service. (Figure 2: The result is a reference to the instance VSMC.)

- 859 2) The client obtains the set of instances of the CIM_SettingsDefineCapabilities association that
860 associate the instance of the CIM_VirtualSystemMigrationCapabilities obtained in step 1) with
861 instances of the CIM_VirtualSystemMigrationSettingData class that describe various details of
862 the capabilities of the virtual system migration service, invoking the intrinsic References()
863 operation with parameter values set as follows:
- 864 – the value of the ObjectName parameter refers to the instance of the
865 CIM_VirtualSystemMigrationCapabilities obtained in step 1) class that represents the
866 virtual system migration service. (Figure 2: A reference to the instance VSMC).
 - 867 – the value of the ResultClass parameter is set to “CIM_SettingsDefineCapabilities”
- 868 The result is a set of instances of the CIM_SettingsDefineCapabilities association that all
869 reference the input instance. (Figure 2: Association instances DEF, P2, P3, MINSSH, MAXSSH
870 and INCSSH).
- 871 3) For each association instance obtained in step 2), the client obtains the referenced instance of
872 the CIM_VirtualSystemMigrationSettingData class that is referenced by the association
873 instance, invoking the intrinsic GetInstance() operation with parameter values set as follows:
- 874 – the value of the InstanceName parameter is set to the value of the PartComponent
875 property taken from respective instance of the CIM_SettingsDefineCapabilities association
876 as obtained in step 2).
- 877 4) For each pair of instances of the CIM_SettingsDefineCapabilities association obtained in step 2)
878 and of the instance of the CIM_VirtualSystemMigrationSettingData class obtained in step 3)
879 (Figure 2: Pairs DEF/VSMD1, P2/VSMD2, P3/VSMD3, MINSSH/MN_SSH, MAXSSH/MX_SSH
880 and INCSSH/IN_SSH), the client inspects the properties values in the instance of the
881 CIM_SettingsDefineState association, as follows:
- 882 – If the value of the ValueRole property is 0 (Default) (Figure 2: Instance DEF), then the
883 reference instance of the CIM_VirtualSystemMigrationSettingData class represents the
884 default settings (Figure 2: Instance VSMD1, indicating that live migration is used as a
885 default migration operation, along with a relative priority of 100, a bandwidth of 10 Megabit
886 per second and that the default transport protocol is "ssh").
 - 887 – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange
888 property is 0 (Point) (Figure 2: Instances P2 and P3), then the referenced instance of the
889 CIM_VirtualSystemMigrationSettingData class represents supported settings (Figure 2:
890 Instances VSMD2 and VSMD3).
 - 891 – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange
892 property is 1 (Minimums) (Figure 2: Instance MINSSH), then the referenced instance of the
893 CIM_VirtualSystemMigrationSettingData class represents minimum supported settings
894 (Figure 2: Instance MN_SSH indicating a minimum bandwidth of 10 megabit per second for
895 the TransportType 2 (SSH). Note that in Figure 2 the association instance MINSSH has set
896 the value 1 (Correlated) for the PropertyPolicy property, indicating that all non-null values
897 are correlated; thus the minimum setting only applies to the TransportType 2 (SSH), but
898 not to other transport types).
 - 899 – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange
900 property is 2 (Maximums) (Figure 2: Instance MAXSSH), then the referenced instance of
901 the CIM_VirtualSystemMigrationSettingData class represents maximum supported settings
902 (Figure 2: Instance MX_SSH, indicating a maximum bandwidth of 10000 megabit per
903 second for the TransportType 2 (SSH).
 - 904 – If the value of the ValueRole property is 3 (Supported) and the value of the ValueRange
905 property is 3 (Increments) (Figure 2: Instance INCSSH), then the referenced instance of
906 the CIM_VirtualSystemMigrationSettingData class represents supported increment for
907 settings (Figure 2: Instance IN_SSH, indicating that the admissible increment for bandwidth
908 is 10 megabit per second).

909 **Result:** The client knows the capabilities of the virtual system migration service, in particular, which virtual
910 system migration types and which transport types are supported, and what are the limitations the apply to
911 the Bandwidth property and the Priority property if used as input instances for methods of the virtual
912 system migration service.

913 9.1.3 Determine life migratability of a virtual system to a target virtualization platform

914 **Assumption:** The client knows all of the following:

- 915 • a reference to an instance of the CIM_ComputerSystem class that represents the virtual system
- 916 • a reference to an instance of the CIM_VirtualSystemMigrationService class representing the
917 responsible virtual system migration service; see 9.1.1
- 918 • the IP address of the target virtualization platform

919 The sequence of activities is as follows:

- 920 1) The client invokes the extrinsic CheckVirtualSystemsIsMigratableToHost() method on the
921 instance of the CIM_VirtualSystemMigrationService, with parameter values set as follows:
 - 922 – the value of the ComputerSystem parameter refers to the instance of the CIM_Computer-
923 System class that represents the virtual system to be migrated
 - 924 – the value of the DestinationHost parameter is set to the IP address of the target
925 virtualization platform
 - 926 – the value of the MigrationSettingData parameter contains an embedded instance of the
927 CIM_VirtualSystemMigrationSettingData class, with property values set as follows:
 - 928 – the value of the MigrationType property is set to 2 (Live)
 - 929 – the value of the Priority property is not set, requesting a default priority
 - 930 – the value of the NewSystemSettingData parameter is set as required
 - 931 – the value of the NewResourceSettingData[] array parameter is set as required to modify
932 virtual resource allocation that existed in the source virtual system or to add new resources
933 into the migrated virtual system
- 934 2) The implementation performs the requested check operation synchronously; the value of output
935 parameters is set as follows:
 - 936 – The value of the return code is 0 (Completed with No Errors)
 - 937 – The value of the IsMigratable property is set to true.

938 **Result:** The client knows that a migration of the virtual system to the target virtualization platform is
939 potentially possible. If the value of the IsMigratable property had been false, that would indicate that a
940 migration as requested would not be possible.

941 **NOTE** The successful execution of the CheckVirtualSystemsIsMigratableToHost() method does not ensure that a
942 subsequent migration operation is successful as conditions such as resource availability may change
943 substantially in short periods of time.

944 9.2 Migration operations

945 This clause lists use-cases describing migration operations of virtual systems.

946 9.2.1 Live migration

947 **Assumption:** All of the following:

- 948 • The client knows a reference to the instance of the CIM_ComputerSystem class that represents
949 the source virtual system

- 950 • The client knows a reference to the instance of the CIM_VirtualSystemMigrationService class
951 that is responsible for the source virtual system; see 9.1.1
- 952 • The client knows a reference to the instance of the CIM_System class that represents the target
953 host system
- 954 • The source virtual system is in a virtual system state that is acceptable by the selected type of
955 migration operation
- 956 • Optionally, the client may pre-check whether the source virtual system is migratable to the
957 target virtualization platform; see 9.1.3. Note though that the migration may still fail.

958 The sequence of activities is as follows:

- 959 1) The client invokes the extrinsic MigrateVirtualSystemToSystem() method on the instance of the
960 CIM_VirtualSystemMigrationService, with parameter values set as follows:
 - 961 – the value of the ComputerSystem parameter refers to the instance of the CIM_Computer-
962 System class that represents the virtual system to be migrated
 - 963 – the value of the DestinationSystem parameter refers to the instance of the CIM_System
964 class that represents the target virtualization platform
 - 965 – the value of the MigrationSettingData parameter contains an embedded instance of the
966 CIM_VirtualSystemMigrationSettingData class, with property values set as follows:
 - 967 – the value of the MigrationType property is set to 2 (Live)
 - 968 – the value of the Priority property is not set, requesting a default priority
 - 969 – the value of the NewSystemSettingData parameter is not set
 - 970 – the value of the NewResourceSettingData[] array parameter is not set
- 971 2) The implementation initiates the requested operation as an asynchronous task; the value of
972 output parameters is set as follows:
 - 973 – The value of the return code is 4096 (Method Parameters Checked – Job Started)
 - 974 – The value of the NewComputerSystem parameter is NULL
 - 975 – The value of the Job parameter refers to the instance of the CIM_ConcreteJob class that
976 represents the ongoing migration operation.
- 977 3) The client tracks the state of the ongoing migration operation by repeatedly obtaining the
978 instance of the CIM_ConcreteJob class, invoking the intrinsic GetInstance() CIM operation with
979 parameter values set as follows:
 - 980 – The value of the InstanceName parameter refers to the instance of the CIM_ConcreteJob
981 class using the reference returned in step 2)

982 The result is the instance of the CIM_ConcreteJob class.
- 983 4) The client checks the value of the value of the JobState property in the instance of the
984 CIM_ConcreteJob class.
 - 985 – If the value is one of 2 (New), 3 (Starting), 4 (Running) or 5 (Suspended), the client waits a
986 certain amount of time and then continues repeating step 3).
 - 987 – If the value is 7 (Completed), the client continues with step 5).
 - 988 – If the value is one of 8 (Terminated), 9 (Killed) or 10 (Exception), the client continues with
989 step 6).
 - 990 – The interpretation of other values of the JobState property is undefined by this profile.

- 991 5) The client resolves the CIM_AffectedJobElement association to find the instance of the
992 CIM_ComputerSystem class that represents the migrated virtual system, invoking the intrinsic
993 Associators() CIM operation with parameter values set as follows:
- 994 – The value of the ObjectName parameter refers to the instance of the CIM_ConcreteJob
995 class that represents the completed migration operation.
 - 996 – The value of the AssocClass parameter is set to “CIM_AffectedJobElement”.
 - 997 – The value of the ResultClass parameter is set to “CIM_ComputerSystem”.
- 998 The result of this step is a set of instances of the CIM_ComputerSystem that represent the
999 source virtual system and the target virtual system of the migration operation. From that set, the
1000 client drops the instance that represents the source virtual system, leaving the instance that
1001 represents the target virtual system.
- 1002 The use-case is completed in this case.
- 1003 6) The client invokes the extrinsic GetError() method on the instance of the CIM_ConcreteJob
1004 class. The method has no input parameters.
- 1005 On return, the value of the Error parameters is an embedded instance of the CIM_Error class
1006 that conveys details about the error.
- 1007 **Result:** The migration operation is completed. If the operation completed successfully, the client knows a
1008 reference to the instance of the CIM_ComputerSystem class representing the migrated virtual system. If
1009 the operation failed, the client knows an instance of the CIM_Error class conveying details about the
1010 failure.
- 1011 Figure 3 shows the situation before and after a successful migration operation.

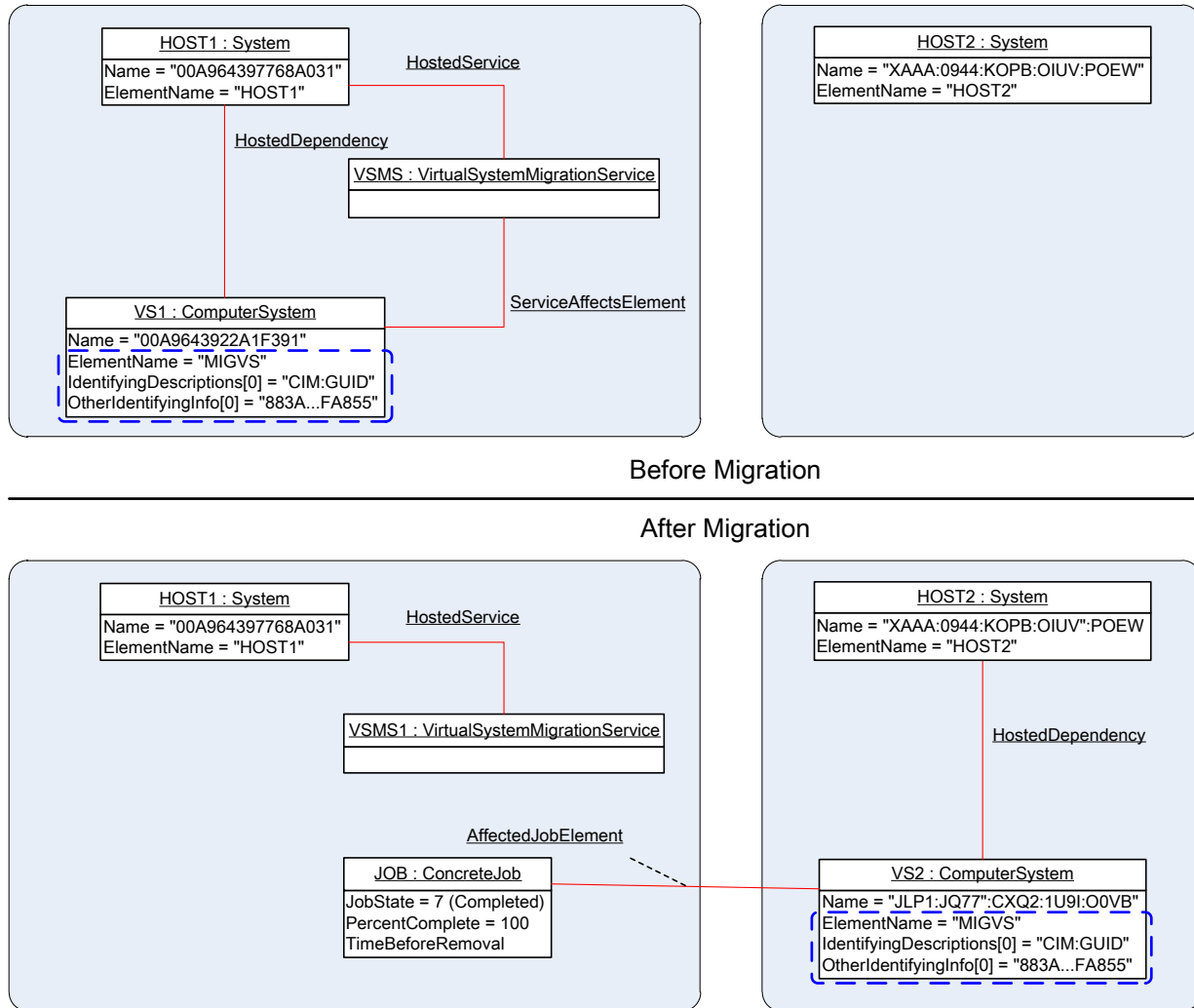


Figure 3 – Instance diagram: Virtual system migration

1012
1013

1014 Note The structure of the values of the Name key property in the instances of the CIM_ComputerSystem class is
1015 implementation dependent. This profile does not specify structure and content of any key properties.
1016 However, if implemented in accordance with [DMTF DSP1052: 1.0.0 \(Computer System Profile\)](#) values of the
1017 IdentifyingDescriptions[] and the OtherIdentifyingInfo[] array properties reflect the fact that the migrated
1018 virtual system is in fact logically identical to the source virtual system by showing identical values for
1019 respective types of correlation properties as defined by [DMTF DSP1052: 1.0.0](#). For example, in Figure 3 the
1020 logical identity is established by means of identical GUIDs by indicating with the value "CIM:GUID" for
1021 IdentifyingDescriptions[0] that GUIDs are used for identification, and providing identical values for the
1022 OtherIdentifyingInfo[0] for both the source and the target virtual system.

1023 **9.2.2 Static migration with request for additional resources**

1024 **Assumption:** All of the following:

- 1025 • The client knows a reference to the instance of the CIM_ComputerSystem class that represents
1026 the source virtual system
- 1027 • The source virtual system has 2 GB of memory defined; this shall be increased to 5 GB as part
1028 of the migration operation
- 1029 • The client knows a reference to the instance of the CIM_VirtualSystemMigrationService class
1030 that is responsible for the source virtual system; see 9.1.1

- 1031 • The client knows a reference to the instance of the CIM_System class that represents the target
1032 host system
- 1033 • The source virtual system is migratable to the target virtualization platform; see 9.1.3

1034 The sequence of activities is as follows:

- 1035 1) The client invokes the extrinsic MigrateVirtualSystemToSystem() method on the instance of the
1036 CIM_VirtualSystemMigrationService, with parameter values set as follows:
 - 1037 – the value of the ComputerSystem parameter refers to the instance of the CIM_Computer-
1038 System class that represents the virtual system to be migrated
 - 1039 – the value of the DestinationSystem parameter refers to the instance of the CIM_System
1040 class that represents the target host system
 - 1041 – the value of the MigrationSettingData parameter contains an embedded instance of the
1042 CIM_VirtualSystemMigrationSettingData class, with property values set as follows:
 - 1043 – the value of the MigrationType property is set to 4 (Restart)
 - 1044 – the value of the Priority property is not set, requesting a default priority
 - 1045 – the value of the NewSystemSettingData parameter is not set
 - 1046 – the value of the NewResourceSettingData[] array parameter is set as follows:
 - 1047 – exactly one array element containing a copy of the instance of the
1048 CIM_ResourceAllocationSettingData class that described the memory allocation with
1049 the definition of the source virtual system; see [DMTF DSP1057](#) (*Virtual System
1050 Profile*) how to obtain that instance
 - 1051 – The value of the AllocationUnits property has a value of “bytes*10^9”, indicating that
1052 memory is allocated in units of Gigabyte (with 1 GB equal to 10^9 byte)
 - 1053 – The value of the Reservation property shall be set to 5, indicating that the amount of
1054 memory in the virtual system definition shall be increased to 5 GB
 - 1055 Note Other combinations of values of the AllocationUnits and the Reservation property may yield
1056 the same result.
- 1057 2) The implementation performs the requested operation as a synchronous task; the value of
1058 output parameters is set as follows:
 - 1059 – The value of the return code is 0 (Completed with No Error)
 - 1060 – The value of the NewComputerSystem parameter refers to the instance of the
1061 CIM_ComputerSystem class representing the migrated virtual system.
 - 1062 – The value of the Job parameter is NULL.

1063 **Result:** The migration operation is completed, the client knows a reference to the instance of the
1064 CIM_ComputerSystem class representing the migrated virtual system. The definition for the virtual
1065 systems memory size is increased to the requested value.

1066 10 CIM elements

1067 Table 17 lists CIM elements that are defined or specialized for this profile. Each CIM element shall be
1068 implemented as described in Table 21. The CIM schema descriptions for any referenced element and its
1069 sub-elements apply.

1070 Clauses 7 (Implementation) and 8 (Methods) may impose additional requirements on these elements.

1071

Table 17 – CIM Elements: Virtual System Migration profile

Element	Requirement	Notes
Classes		
CIM_AffectedJobElement	Conditional	See 10.1
CIM_AssociatedJobMethodResult	Conditional	See 10.2
CIM_ConcreteJob	Conditional	See 10.3
CIM_ComputerSystem	Conditional	See 10.4
CIM_ElementCapabilities	Mandatory	See 10.5
CIM_Error	Conditional	See 10.6
CIM_HostedService	Mandatory	See 10.7
CIM_MethodResult	Conditional	See 10.8
CIM_OwningJobElement	Conditional	See 10.9
CIM_RegisteredProfile	Mandatory	See 10.10
CIM_ServiceAffectsElement	Mandatory	See 10.11
CIM_SettingsDefineCapabilities	Mandatory	See 10.12
CIM_VirtualSystemMigrationCapabilities	Mandatory	See 10.13
CIM_VirtualSystemMigrationService	Mandatory	See 10.14
CIM_VirtualSystemMigrationSettingData (Parameter)	Mandatory	See 10.15
CIM_VirtualSystemMigrationSettingData (Capabilities)	Mandatory	See 10.16
Indications		
Select * CIM_InstCreation Where SourceInstance ISA CIM_ConcreteJob	Conditional	Query Language: CQL Lifecycle indication representing the VS_MIGRATION_JOB_CREATE event See 10.17
Select * CIM_InstDeletion Where SourceInstance ISA CIM_ConcreteJob	Conditional	Query Language: CQL. Lifecycle indication representing the VS_MIGRATION_JOB_DELETE event See 10.18
Select * CIM_InstMethodCall Where SourceInstance ISA CIM_VirtualSystemMigrationService	Conditional	Query Language: CQL. Lifecycle indication representing the VS_MIGRATION_METHOD_CALL event. See 10.19
Select * CIM_InstModification Where SourceInstance ISA CIM_ConcreteJob AND PreviousInstance ISA CIM_ConcreteJob AND SourceInstance <> PreviousInstance	Conditional	Query Language: CQL. Lifecycle indication representing the VS_MIGRATION_JOB_CHANGE event. See 10.20

1072 **10.1 CIM_AffectedJobElement**

1073 The implementation of the CIM_AffectedJobElement association is conditional.

1074 Condition: The CIM_AffectedJobElement association shall be implemented if one or more of the methods
 1075 of the CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior;
 1076 see 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1077 The CIM_AffectedJobElement association shall be instantiated between the instance of the
 1078 CIM_ConcreteJob class representing a virtual system migration task and instances of the
 1079 CIM_ComputerSystem class representing the source and the target virtual systems that are affected by
 1080 the virtual system migration task.

1081 Table 18 lists the requirements for elements of this association. These requirements are in addition to
 1082 those specified in the CIM Schema and – if implemented – in (*Job Control Profile*).

1083 **Table 18 – Association: CIM_AffectedJobElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	Key: Value shall reference the instance of the CIM_ComputerSystem class Cardinality: 1..2
AffectingElement	Mandatory	Key: Value shall reference the instance of the CIM_ConcreteJob class Cardinality: 1
ElementEffects[]	Mandatory	Value shall have exactly one element that has the value 1 (Other).
OtherElementEffectsDescription[]	Mandatory	Value shall have exactly one element that has the value "Virtual System Migration".

1084 10.2 CIM_AssociatedJobMethodResult

1085 The implementation of the CIM_AssociatedJobMethodResult association is conditional.

1086 Condition: The CIM_AssociatedJobMethodResult association shall be implemented if one or more of the
 1087 methods of the CIM_VirtualSystemMigrationService class is implemented with asynchronous execution
 1088 behavior; see 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1089 The CIM_AssociatedJobMethodResult association shall be instantiated between the instance of the
 1090 CIM_ConcreteJob class representing a virtual system migration task and the instance of the
 1091 CIM_MethodResult class representing the related method invocation.

1092 Table 19 lists the requirements for elements of this association. These requirements are in addition to
 1093 those specified in the CIM Schema, and – if implemented - in (*Job Control Profile*).

1094 **Table 19 – Association: CIM_AssociatedJobMethodResult**

Elements	Requirement	Notes
Job	Mandatory	Key: Value shall reference the instance of the CIM_Concrete job class Cardinality: 1
JobParameters	Mandatory	Key: Value shall reference the instance of the CIM_MethodResult class Cardinality: 1

1095 10.3 CIM_ConcreteJob

1096 The implementation of the CIM_ConcreteJob class is conditional.

1097 Condition: The CIM_ConcreteJob class shall be implemented if one or more of the methods of the
 1098 CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior; see
 1099 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1100 Instances of the CIM_ConcreteJob class shall represent asynchronous virtual system migration tasks.

1101 Table 20 lists the requirements for elements of this class. These requirements are in addition to those
 1102 specified in the CIM Schema and - if implemented - in (*Job Control Profile*).

1103 **Table 20 – Class: CIM_ConcreteJob**

Element	Requirement	Description
JobState	Mandatory	See CIM Schema
TimeOfLastStateChange	Mandatory	See CIM Schema

1104 **10.4 CIM_ComputerSystem**

1105 The implementation of the CIM_ComputerSystem class is conditional.

1106 Condition: The implementation of the CIM_ComputerSystem class shall be further constrained as
 1107 specified in this clause beyond the requirements specified in [DMTF DSP1057: 1.0.0](#) (*Virtual System*
 1108 *Profile*), Subclause 10.2, and in [DMTF DSP1052: 1.0.0](#) (*Computer System Profile*), Subclause 10.1 if the
 1109 optional array properties OtherIdentifyingInfo[] and IdentifyingDescriptions[] array properties as specified
 1110 by the [DMTF DSP1052: 1.0.0](#), Subclause 10.1 for instances of the CIM_ComputerSystem class
 1111 representing virtual systems are implemented.

1112 Instances of the CIM_ComputerSystem class shall be used for the representation of the source and the
 1113 target virtual system of a virtual system migration process.

1114 Table 21 lists the requirements for elements of this class.

1115 **Table 21 – Class: CIM_ComputerSystem**

Element	Requirement	Description
ElementName	Optional	See 7.5
IdentifyingDescriptions[]	Conditional	See 7.5.2
OtherIdentifyingInfo[]	Conditional	See 7.5.3

1116 **10.5 CIM_ElementCapabilities**

1117 The CIM_ElementCapabilities association shall be instantiated between an instance of the
 1118 CIM_VirtualSystemMigrationService class representing a virtual system migration service and an instance
 1119 of the CIM_VirtualSystemMigrationCapabilities class representing the capabilities of that virtual system
 1120 migration service.

1121 Table 22 lists the requirements for elements of this association.

1122 **Table 22 – Association: CIM_ElementCapabilities**

Element	Requirement	Notes
ManagedElement	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationService class Cardinality: *
Capabilities	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationCapabilities class Cardinality: *

1123 10.6 CIM_Error

1124 The implementation of the CIM_Error class is conditional.

1125 Condition: The CIM_Error class shall be implemented if standard messages are implemented; see 8.1.1,
1126 8.1.2, 8.1.3, 8.1.4 and subclauses of 8.2.

1127 Instances of the CIM_Error class shall be used to convey detailed error information if the execution of
1128 extrinsic methods or of intrinsic generic operations fails.

1129 Table 23 lists the requirements for elements of this class.

1130 **Table 23 – Class: CIM_Error**

Element	Requirement	Notes
MessageID	Mandatory	Value shall identify the standard message conveyed through the instance of the CIM_Error class.
Message	Mandatory	Value shall contain the formatted standard message.
MessageArguments	Mandatory	Value shall contain the dynamic content of the message.
ErrorSource	Conditional	Value shall identify the primary entity that caused the error condition.

1131 10.7 CIM_HostedService

1132 The CIM_HostedService association shall be instantiated between the instance of the CIM_System class
1133 representing a host system and the instance of the CIM_VirtualSystemMigrationService class
1134 representing a virtual system migration service.

1135 Table 24 lists the requirements for elements of this association.

1136

Table 24 – Association: CIM_HostedService

Elements	Requirement	Notes
Antecedent	Mandatory	Key: Value shall reference the instance of the CIM_System class Cardinality: 1
Dependent	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationService class Cardinality: 1

1137 **10.8 CIM_MethodResult**

1138 The implementation of the CIM_MethodResult class is conditional.

1139 Condition: The CIM_MethodResult class shall be implemented if one or more of the methods of the
1140 CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior; see
1141 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1142 The CIM_MethodResult class shall represent invocations of methods of the
1143 CIM_VirtualSystemMigrationService that are executed asynchronously.

1144 Table 25 lists the requirements for elements of this class. These requirements are in addition to those
1145 specified in the CIM Schema, and - if implemented – in (*Job Control Profile*).

1146

Table 25 – Class: CIM_MethodResult

Element	Requirement	Notes
InstanceID	Mandatory	Key
PreCallIndication	Mandatory	Value shall be an embedded instance of the CIM_InstMethodCall indication representing pre-execution values
PostCallIndication	Conditional	If method execution is complete, value shall be an embedded instance of the CIM_InstMethodCall indication representing post-execution values

1147 **10.9 CIM_OwningJobElement**

1148 The implementation of the CIM_OwningJobElement association is conditional.

1149 Condition: The CIM_OwningJobElement association shall be implemented if one or more of the methods
1150 of the CIM_VirtualSystemMigrationService class is implemented with asynchronous execution behavior;
1151 see 8.1.1, 8.1.2, 8.1.3 and 8.1.4.

1152 The CIM_OwningJobElement association shall be instantiated between the instance of the
1153 CIM_VirtualSystemMigrationService class representing a virtual system migration service and the
1154 instance of the CIM_ConcreteJob class representing a virtual system migration task that is owned by the
1155 service.

1156 Table 26 lists requirements for elements of this association. These requirements are in addition to those
1157 specified in the CIM Schema and - if implemented - in (*Job Control Profile*).

1158

Table 26 – Association: CIM_OwningJobElement

Elements	Requirement	Notes
OwningElement	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationService class Cardinality: 1
OwnedElement	Mandatory	Key: Value shall reference the instance of the CIM_ConcreteJob Cardinality: *

1159 10.10 CIM_RegisteredProfile

1160 Table 27 lists the requirements for elements of this class. These requirements are in addition to those
1161 specified by [DMTF DSP1033:1.0.0 \(Profile Registration Profile\)](#).

1162

Table 27 – Class: CIM_RegisteredProfile

Elements	Requirement	Notes
RegisteredOrganization	Mandatory	Value shall be set to 2 (DMTF)
RegisteredName	Mandatory	Value shall be set to "Virtual System Migration"
RegisteredVersion	Mandatory	Value shall be set to the version of this profile: "1.0.0".

1163 10.11 CIM_ServiceAffectsElement

1164 The CIM_ServiceAffectsElement association shall be instantiated between instances of the
1165 CIM_ComputerSystem class representing a virtual system and an instance of the
1166 CIM_VirtualSystemMigrationService class representing a virtual system migration service that is capable
1167 of managing migration operations for the virtual system.

1168 Table 28 lists the requirements for elements of this association.

1169

Table 28 – Association: CIM_ServiceAffectsElement

Element	Requirement	Notes
AffectedElement	Mandatory	Key: Value shall reference the instance of the CIM_ComputerSystem class representing the virtual system Cardinality: *
AffectingElement	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationService class representing the managing virtual system migration service Cardinality: 1
ElementEffects[]	Mandatory	Value shall have exactly one element with the value set to 5 (Manages)
AssignedSequence	Mandatory	Value shall reflect the requested priority of a requested (pending) or ongoing migration operation

1170 **10.12 CIM_SettingsDefineCapabilities**

1171 The CIM_SettingsDefineCapabilities association shall be instantiated between an instance of the
 1172 CIM_VirtualSystemMigrationCapabilities class representing capabilities of a virtual system migration
 1173 service and an instance of the CIM_VirtualSystemMigrationSettingData class representing default values
 1174 for operations of the service.

1175 Table 29 lists the requirements for elements of this association.

1176 **Table 29 – Association: CIM_SettingsDefineCapabilities**

Element	Requirement	Notes
GroupComponent	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationCapabilities class representing capabilities of a virtual system migration service. Cardinality: 1
PartComponent	Mandatory	Key: Value shall reference the instance of the CIM_VirtualSystemMigrationSettingData class representing default migration settings Cardinality: 1..*
PropertyPolicy	Mandatory	See 7.4.1 for default migration settings and 7.4.2 for admissible migration settings.
ValueRole	Mandatory	See 7.4.1 for default migration settings and 7.4.2 for admissible migration settings.
ValueRange	Mandatory	See 7.4.1 for default migration settings and 7.4.2 for admissible migration settings.

1177 **10.13 CIM_VirtualSystemMigrationCapabilities**

1178 The CIM_VirtualSystemMigrationCapabilities class represents capabilities of one or more virtual system
 1179 migration services.

1180 Table 30 lists the requirements for elements of this class.

1181 **Table 30 – Class: CIM_VirtualSystemMigrationCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
AsynchronousMethodsSupported[]	Mandatory	See 7.2.2
SynchronousMethodsSupported[]	Mandatory	See 7.2.2
DestinationHostsFormatsSupported[]	Conditional	Required if the optional CheckVirtualSystemIsMigratableToHost() method (see 8.1.2) and/or the optional MigrateVirtualSystemToHost() method (see 8.1.4) are implemented.

1182 **10.14 CIM_VirtualSystemMigrationService**

1183 The CIM_VirtualSystemMigrationService class models a virtual system migration service.

1184 Instances of the CIM_VirtualSystemMigrationService shall represent virtual system migration services.

1185 Table 31 lists the requirements for elements of this class.

1186 **Table 31 – Class: CIM_VirtualSystemMigrationService**

Elements	Requirement	Notes
CreationClassName	Mandatory	Key
Name	Mandatory	Key
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
MigrateVirtualSystemToHost()	Conditional	See 8.1.1
MigrateVirtualSystemToSystem()	Optional	See 8.1.2
CheckVirtualSystemIsMigratableToHost()	Conditional	See 8.1.3
CheckVirtualSystemIsMigratableToSystem()	Optional	See 8.1.4

1187

1188 10.15 CIM_VirtualSystemMigrationSettingData (Parameter)

1189 Instances of the CIM_VirtualSystemMigrationSettingData class shall represent the parameterization of
1190 operations of the CIM_VirtualSystemMigrationService class.

1191 Table 32 lists the requirements for elements of this class if used as a parameter of methods of the
1192 CIM_VirtualSystemMigrationService class.

1193 **Table 32 – Class: CIM_VirtualSystemMigrationSettingData (Parameter)**

Elements	Requirement	Notes
InstanceID	Mandatory	Key: Shall be NULL
MigrationType	Optional	See 7.3.1
Priority	Optional	See 7.3.2
Bandwidth	Optional	See 7.3.3
BandwidthUnit	Optional	See 7.3.4
TransportType	Optional	See 7.3.5
OtherTransportType	Conditional	See 7.3.6

1194 10.16 CIM_VirtualSystemMigrationSettingData (Capabilities)

1195 Instances of the CIM_VirtualSystemMigrationSettingData class shall represent default values that apply
1196 to method invocations of the CIM_VirtualSystemMigrationService class. Instances of the
1197 CIM_VirtualSystemMigrationSettingData class may represent admissible values for method invocations of
1198 the CIM_VirtualSystemMigrationService class if referenced by instances of the
1199 CIM_SettingsDefineCapabilities association where the value of the ValueRole property is either 0
1200 (Default) or 3 (Supported).

1201 Table 33 contains the requirements for instances of this class if used as default value expressing
 1202 capabilities of the CIM_VirtualSystemMigrationService class.

1203 **Table 33 – Class: CIM_VirtualSystemMigrationSettingData (Capabilities)**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
MigrationType	Mandatory	See 7.4.3
Priority	Mandatory	See 7.4.4
Bandwidth	Optional	See 7.4.5
BandwidthUnit	Conditional	See 7.4.6
TransportType	Optional	See 7.4.7
OtherTransportType	Conditional	See 7.4.8

1204

1205 **10.17 CIM_InstCreation**

1206 The implementation of the CIM_InstCreation indication is conditional.

1207 Condition: The CIM_InstCreation indication shall be implemented if indications (see 7.6) are implemented
 1208 reporting the VS_MIGRATION_JOB_CREATE event.

1209 Table 34 contains the requirements for this case. These requirements are in addition to those specified in
 1210 the CIM Schema and in the *Indications Profile*.

1211 **Table 34 – Indication: CIM_InstCreation**

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[]	Conditional	See 7.6
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_CREATE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_ConcreteJob class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system

1212 **10.18 CIM_InstDeletion**

1213 The implementation of the CIM_InstDeletion indication is conditional.

1214 Condition: The CIM_InstDeletion indication shall be implemented if any indications (see 7.6) are
 1215 implemented reporting the VS_MIGRATION_JOB_DELETE event.

1216 Table 35 contains the requirements for this case. These requirements are in addition to those specified in
 1217 the CIM Schema and in [DMTF DSP1054](#) (*Indications Profile*).

1218 **Table 35 – Indication: CIM_InstDeletion**

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[]	Conditional	See 7.6
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_DELETE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_ConcreteJob class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system

1219

1220 **10.19 CIM_InstMethodCall**

1221 The implementation of the CIM_InstMethodCall indication is conditional.

1222 Condition: The CIM_InstMethodCall indication shall be implemented if any indications (see 7.6) are
 1223 implemented reporting the VS_MIGRATION_METHOD_CALL event.

1224 Table 36 contains the requirements for this case. These requirements are in addition to those specified in
 1225 the CIM Schema and in [DMTF DSP1054](#) (*Indications Profile*).

1226

Table 36 – Indication: CIM_InstMethodCall

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[]	Mandatory	Unspecified
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_CHANGE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_VirtualSystemMigrationService class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system
MethodName	Mandatory	See 7.6
MethodParameters	Mandatory	See 7.6
ReturnValue	Conditional	Condition: Value of the PreCall property is FALSE - For value definition see CIM Schema
ReturnValueType	Conditional	Condition: Value of the PreCall property is FALSE - For value definition see CIM Schema
Error[]	Optional	See CIM Schema
PreCall	Mandatory	See CIM Schema

1227 **10.20 CIM_InstModification**

1228 The implementation of the CIM_InstModification indication is conditional.

1229 Condition: The CIM_InstModification indication shall be implemented if any indications (see 7.6) are
 1230 implemented reporting the VS_MIGRATION_JOB_CHANGE event.

1231 Table 37 contains the requirements for this case. These requirements are in addition to those specified in
 1232 the CIM Schema and in [DMTF DSP1054](#) (*Indications Profile*).

1233

Table 37 – Indication: CIM_InstModification

Elements	Requirement	Notes
IndicationIdentifier	Mandatory	Value shall contain a unique identification of the indication instance; for the format see CIM Schema definition.
CorrelatedIndications[]	Mandatory	Unspecified
IndicationTime	Mandatory	Value shall contain the time of the reported VS_MIGRATION_JOB_CHANGE event
PerceivedSeverity	Mandatory	Value shall be 2 (Information)
IndicationFilterName	Mandatory	See 7.6

Elements	Requirement	Notes
SourceInstance	Mandatory	See 7.6
SourceInstanceModelPath	Mandatory	Value shall refer to the instance of the CIM_ConcreteJob class copied into the SourceInstance parameter
SourceInstanceHost	Optional	Value should contain the host name or IP address of the source host system
PreviousInstance	Optional	See 7.6

1234

**ANNEX A
(informative)****Change log**1235
1236
1237
1238

1239

Version	Date	Description
1.0.0	2012-08-21	

1240