



1

2

3

4

Document Number: DSP1020

Date: 2009-06-16

Version: 1.0.0

5 **Pass-Through Module Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9

10 Copyright Notice

11 Copyright © 2006, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

33	Foreword	5
34	Introduction	6
35	1 Scope	7
36	2 Normative References.....	7
37	2.1 Approved References	7
38	2.2 Other References.....	7
39	3 Terms and Definitions	7
40	4 Symbols and Abbreviated Terms	8
41	5 Synopsis	9
42	6 Description	9
43	7 Implementation Requirements	10
44	7.1 CIM_PassThroughModule	10
45	7.2 Management of Port Assignments.....	12
46	8 Methods.....	13
47	8.1 Method: CIM_PassThroughModule.AssignPorts()	13
48	8.2 Method: CIM_PassThroughModule.RequestStateChange()	13
49	8.3 Profile Conventions for Operations.....	14
50	8.4 CIM_ElementCapabilities	14
51	8.5 CIM_EnabledLogicalElementCapabilities.....	15
52	8.6 CIM_PassThroughModule	15
53	8.7 CIM_SystemDevice	15
54	9 Use Cases	16
55	9.1 Object Diagrams	16
56	9.2 Determine Pass-Through Module Link Technology	17
57	9.3 Determine Pass-Through Module Port Mappings	17
58	9.4 Determine Whether Port Mappings Are Configurable	17
59	9.5 Manage Port Mappings on a Pass-Through Module	17
60	9.6 Determining If ElementName Can Be Modified.....	18
61	9.7 Determining If State Management Is Supported	18
62	10 CIM Elements	18
63	10.1 CIM_ElementCapabilities	19
64	10.2 CIM_EnabledLogicalElementCapabilities.....	19
65	10.3 CIM_PassThroughModule	20
66	10.4 CIM_RegisteredProfile.....	20
67	10.5 CIM_SystemDevice	21
68	ANNEX A (informative) Change Log.....	22
69		

70 **Figures**

71	Figure 1 – Pass-Through Module Profile: Class Diagram	10
72	Figure 2 – Instance Diagram.....	16
73	Figure 3 – Port Mappings Crossed	17
74		

75 **Tables**

76	Table 1 – Referenced Profiles	9
77	Table 2 – CIM_PassThroughModule.AssignPorts() Method: Return Code Values.....	13
78	Table 3 – CIM_PassThroughModule.AssignPorts() Method: Parameters.....	13
79	Table 4 – CIM_PassThroughModule.RequestStateChange() Method: Return Code Values	14
80	Table 5 – CIM_PassThroughModule.RequestStateChange() Method: Parameters	14
81	Table 6 – Operations: CIM_ElementCapabilities	15
82	Table 7 – Operations: CIM_PassThroughModule	15
83	Table 8 – Operations: CIM_SystemDevice.....	16
84	Table 9 – Required CIM Elements: Pass-Through Module Profile.....	18
85	Table 10 – Class: CIM_ElementCapabilities.....	19
86	Table 11 – Class: CIM_EnabledLogicalElementCapabilities.....	19
87	Table 12 – Class: CIM_PassThroughModule	20
88	Table 13 – Class: CIM_RegisteredProfile.....	20
89	Table 14 – Class: CIM_SystemDevice	21

90

91

Foreword

92 The *Pass-Through Module Profile* (DSP1020) was prepared by the Server Management Working Group
93 and the Physical Platform Profiles Working Group.

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

96

Acknowledgments

97 The authors wish to acknowledge the following people.

98 **Editor:**

- 99
 - Aaron Merkin – IBM

100 **Contributors:**

- 101
 - Jon Hass – Dell
- 102
 - Khachatur Papanyan – Dell
- 103
 - Enoch Suen – Dell
- 104
 - Jeff Hilland – HP
- 105
 - Christina Shaw – HP
- 106
 - Aaron Merkin – IBM
- 107
 - Perry Vincent – Intel
- 108
 - John Leung – Intel

109

Introduction

110 The information in this specification should be sufficient for a provider or consumer of this data to identify
111 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
112 represent and manage a pass-through module of a modular system that is modeled using the DMTF CIM
113 core and extended model definitions.

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

116

Pass-Through Module Profile

117 1 Scope

118 The *Pass-Through Module Profile* is a component profile for modeling pass-through modules of modular
119 systems. A pass-through module acts as a conduit for network connectivity for components within a
120 modular system without performing any higher order network protocol function.

121 2 Normative References

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

125 2.1 Approved References

126 DMTF DSP0004, *CIM Infrastructure Specification 2.3*,
127 http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

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

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

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

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

136 2.2 Other References

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

139 3 Terms and Definitions

140 For the purposes of this document, the following terms and definitions apply.

141 3.1

142 **can**

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

144 3.2

145 **cannot**

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

147 3.3

148 **conditional**

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

- 151 **3.4**
152 **mandatory**
153 indicates requirements to be followed strictly in order to conform to the document and from which no
154 deviation is permitted
- 155 **3.5**
156 **may**
157 indicates a course of action permissible within the limits of the document
- 158 **3.6**
159 **need not**
160 indicates a course of action permissible within the limits of the document
- 161 **3.7**
162 **optional**
163 indicates a course of action permissible within the limits of the document
- 164 **3.8**
165 **referencing profile**
166 indicates a profile that owns the definition of this class and can include a reference to this profile in its
167 "Referenced Profiles" table
- 168 **3.9**
169 **shall**
170 indicates requirements to be followed strictly in order to conform to the document and from which no
171 deviation is permitted
- 172 **3.10**
173 **shall not**
174 indicates requirements to be followed strictly in order to conform to the document and from which no
175 deviation is permitted
- 176 **3.11**
177 **should**
178 indicates that among several possibilities, one is recommended as particularly suitable, without
179 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 180 **3.12**
181 **should not**
182 indicates that a certain possibility or course of action is deprecated but not prohibited

183 **4 Symbols and Abbreviated Terms**

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

- 185 **4.1**
186 **CIM**
187 Common Information Model

188 5 Synopsis

189 **Profile Name:** Pass-Through Module

190 **Version:** 1.0.0

191 **Organization:** DMTF

192 **CIM Schema Version:** 2.22

193 **Central Class:** CIM_PassThroughModule

194 **Scoping Class:** CIM_ComputerSystem

195 The *Pass-Through Module Profile* extends management capability to include support for pass-through
196 modules of modular systems.

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

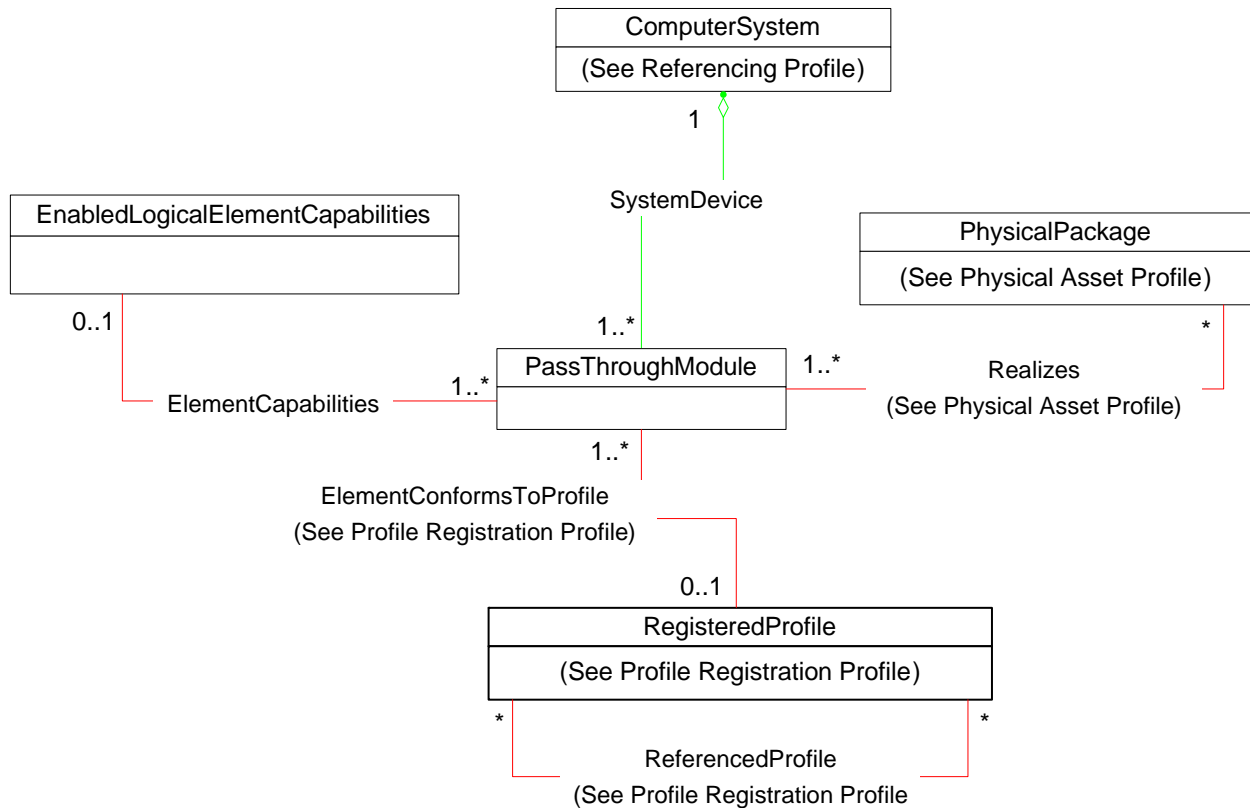
198 **Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Description
Profile Registration	DMTF	1.0	Mandatory
Physical Asset	DMTF	1.0	Optional

199 6 Description

200 The *Pass-Through Module Profile* describes pass-through modules of modular systems. A pass-through
201 module is a device that is a replacement for physical cables and allows internal network physical ports in
202 a chassis or rack to be accessible from the external network. A pass-through module may be a fixed
203 internal-port-to-external-port relationship or a configurable mapping of internal ports to output ports
204 through a cross-point switching function.

205 Figure 1 represents the class schema for the *Pass-Through Module Profile*. For simplicity, the prefix CIM_
 206 has been removed from the names of the classes.



207

208

Figure 1 – Pass-Through Module Profile: Class Diagram

209 7 Implementation Requirements

210 This section details the requirements related to the arrangement of instances and their properties for
 211 implementations of this profile. Required methods are listed in section 8, and properties are listed in
 212 section 10.

213 7.1 CIM_PassThroughModule

214 An instance of CIM_PassThroughModule shall represent the pass-through module.

215 7.1.1 Pass-Through Module State Management Is Supported—Conditional

216 When management of the state of a pass-through module is supported, exactly one instance of
 217 CIM_EnabledLogicalElementCapabilities shall be associated with the CIM_PassThroughModule instance
 218 through an instance of CIM_ElementCapabilities.

219 Support for managing the state of the pass-through module is optional behavior. This section describes
 220 the CIM elements and behaviors that shall be implemented when this behavior is supported.

221 **7.1.1.1 CIM_EnabledLogicalElementCapabilities**

222 When state management is supported, exactly one instance of CIM_EnabledLogicalElement capabilities
223 shall be associated with the CIM_PassThroughModule instance through an instance of the
224 CIM_ElementCapabilities association.

225 **7.1.1.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

226 The RequestedStatesSupported property may contain zero or more of the following values: 2 (Enabled),
227 3 (Disabled), or 11 (Reset).

228 **7.1.1.2 CIM_PassThroughModule.RequestedState**

229 When the CIM_PassThroughModule.RequestStateChange() method is successfully invoked, the value of
230 the RequestedState property shall be the value of the RequestedState parameter. If the method is not
231 successfully invoked, the value of the RequestedState property is indeterminate.

232 The CIM_PassThroughModule.RequestedState property shall have one of the values specified in the
233 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property or 5 (No Change).

234 **7.1.1.3 CIM_PassThroughModule.EnabledState**

235 When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled) and the
236 CIM_PassThroughModule.RequestStateChange() method completes successfully, the value of the
237 EnabledState property shall equal the value of the CIM_PassThroughModule.RequestedState property.

238 If the method does not complete successfully, the value of the EnabledState property is indeterminate.

239 The EnabledState property shall have the value 2 (Enabled), 3 (Disabled), or 5 (Not Applicable).

240 **7.1.2 Pass-Through Module State Management Is Not Supported**

241 This section describes the CIM elements and behaviors that shall be implemented when management of
242 the Pass-Through Module state is not supported.

243 **7.1.2.1 CIM_EnabledLogicalElementCapabilities**

244 When state management is not supported, exactly one instance of
245 CIM_EnabledLogicalElementCapabilities may be associated with the CIM_PassThroughModule instance
246 through an instance of the CIM_ElementCapabilities association.

247 **7.1.2.1.1 CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported**

248 The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property shall not contain any
249 values.

250 **7.1.2.2 CIM_PassThroughModule.RequestedState**

251 The RequestedState property shall have the value 12 (Not Applicable).

252 **7.1.2.3 CIM_PassThroughModule.EnabledState**

253 The EnabledState property shall have one of the following values: 2 (Enabled), 3 (Disabled), or 5 (Not
254 Applicable).

255 **7.1.3 Modifying ElementName Is Supported—Conditional**

256 The CIM_PassThroughModule.ElementName property may support being modified by the ModifyInstance
257 operation. See section 8.6.1.1. This behavior is conditional. This section describes the CIM elements and
258 behavior requirements when an implementation supports client modification of the
259 CIM_PassThroughModule.ElementName property.

260 **7.1.3.1 CIM_EnabledLogicalElementCapabilities**

261 An instance of CIM_EnabledLogicalElementCapabilities shall be associated with the
262 CIM_PassThroughModule instance through an instance of CIM_ElementCapabilities.

263 **7.1.3.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

264 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
265 client modification of the CIM_PassThroughModule.ElementName property.

266 **7.1.3.1.2 CIM_EnabledLogicalElement.MaxElementNameLen**

267 The MaxElementNameLen property shall be implemented.

268 **7.1.4 Modifying ElementName Is Not Supported**

269 This section describes the CIM elements and behaviors that shall be implemented when the
270 CIM_PassThroughModule.ElementName property does not support being modified by the ModifyInstance
271 operation.

272 **7.1.4.1 CIM_EnabledLogicalElementCapabilities**

273 An instance of CIM_EnabledLogicalElementCapabilities may be associated with the
274 CIM_PassThroughModule instance through an instance of CIM_ElementCapabilities.

275 **7.1.4.1.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

276 The ElementNameEditSupported property shall have a value of FALSE when the implementation does
277 not support client modification of the CIM_PassThroughModule.ElementName property.

278 **7.1.4.1.2 CIM_EnabledLogicalElement.MaxElementNameLen**

279 The MaxElementNameLen property may be implemented. The MaxElementNameLen property is
280 irrelevant in this context.

281 **7.2 Management of Port Assignments**

282 An implementation may support management of port assignments.

283 **7.2.1 CIM_PassThroughModule.IsProgrammable**

284 When an implementation supports management of port assignments, the IsProgrammable property of the
285 CIM_PassThroughModule instance shall have a value of TRUE.

286 **7.2.2 Mapping Ports through the AssignPorts Method**

287 Support for mapping ports on the CIM_PassThroughModule instance through the AssignPorts() method
288 is conditional behavior. When the IsProgrammable property has a value of TRUE, the AssignPorts()
289 method shall be implemented and shall be supported.

290 8 Methods

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

293 8.1 Method: CIM_PassThroughModule.AssignPorts()

294 Invocation of the CIM_PassThroughModule.AssignPorts() method creates a mapping or removes a
295 mapping between an internal and external port pair.

296 Detailed requirements of the AssignPorts() method are specified in Table 2 and Table 3.

297 No standard messages are defined.

298 **Table 2 – CIM_PassThroughModule.AssignPorts() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred

299 **Table 3 – CIM_PassThroughModule.AssignPorts() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Mapped	Boolean	If TRUE, the ports will be mapped to each other. If FALSE, the ports will be unmapped from each other.
IN	InternalPort	uint16	Identifies the internal port to be mapped
IN	ExternalPort	uint16	Identifies the external port to be mapped

300 8.2 Method: CIM_PassThroughModule.RequestStateChange()

301 Invocation of the CIM_PassThroughModule.RequestStateChange() method changes the element's state
302 to the value specified in the RequestedState parameter. The 2 (Enabled) and 3 (Disabled) values of the
303 RequestedState parameter shall correspond to enabling or disabling the module represented by the
304 instance of CIM_PassThroughModule on or off accordingly.

305 See section 7.1.1.2 for information about the effect of this method on the RequestedState property.

306 The method shall be considered successful if the availability of the module upon completion of the
307 method corresponds to the desired availability indicated by the RequestedState parameter. An actual
308 change in state is not necessary for the method to be considered successful as long as the resultant state
309 is equal to the requested state. Upon successful completion of the method, the Return Value shall be 0
310 (zero).

311 See section 7.1.2.3 for information about the effect of this method on the EnabledState property.

312 Detailed requirements of the RequestStateChange() method are specified in Table 4 and Table 5.

313 No standard messages are defined.

314 Invoking the CIM_PassThroughModule.RequestStateChange() method multiple times could result in
315 earlier requests being overwritten or lost.

316 **Table 4 – CIM_PassThroughModule.RequestStateChange() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
2	Error occurred
0x1000	Job started: REF returned to started CIM_ConcreteJob

317 **Table 5 – CIM_PassThroughModule.RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values: 2 (Enabled) 3 (Disabled) 11 (Reset)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN, REQ	TimeoutPeriod	datetime	Client specified maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

318 **8.3 Profile Conventions for Operations**

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

321 The default list of operations is as follows:

- 322 • GetInstance
- 323 • Associators
- 324 • AssociatorNames
- 325 • References
- 326 • ReferenceNames
- 327 • EnumerateInstances
- 328 • EnumerateInstanceNames

329 **8.4 CIM_ElementCapabilities**

330 Table 6 lists implementation requirements for operations. If implemented, these operations shall be
331 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in
332 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

334

Table 6 – Operations: CIM_ElementCapabilities

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

335 8.5 CIM_EnabledLogicalElementCapabilities

336 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

338 8.6 CIM_PassThroughModule

339 Table 7 lists implementation requirements for operations. If implemented, these operations shall be
 340 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in
 341 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

343

Table 7 – Operations: CIM_PassThroughModule

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.6.1.	None

344 8.6.1 CIM_PassThroughModule—ModifyInstance Operation

345 This section details the specific requirements for the ModifyInstance operation that is applied to an
 346 instance of CIM_PassThroughModule.

347 8.6.1.1 CIM_PassThroughModule.ElementName Property

348 When an instance of CIM_EnabledLogicalElementCapabilities is associated with the
 349 CIM_PassThroughModule instance and the
 350 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported property has a value of TRUE, the
 351 implementation shall allow the ModifyInstance operation to change the value of the ElementName
 352 property of the CIM_PassThroughModule instance. The ModifyInstance operation shall enforce the length
 353 restriction specified in the MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities
 354 instance.

355 When an instance of CIM_EnabledLogicalElementCapabilities is not associated with the
 356 CIM_PassThroughModule instance, or the ElementNameEditSupported property of the
 357 CIM_EnabledLogicalElementCapabilities has a value of FALSE, the implementation shall not allow the
 358 ModifyInstance operation to change the value of the ElementName property of the
 359 CIM_PassThroughModule instance.

360 8.7 CIM_SystemDevice

361 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
 362 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in
 363 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

365

Table 8 – Operations: CIM_SystemDevice

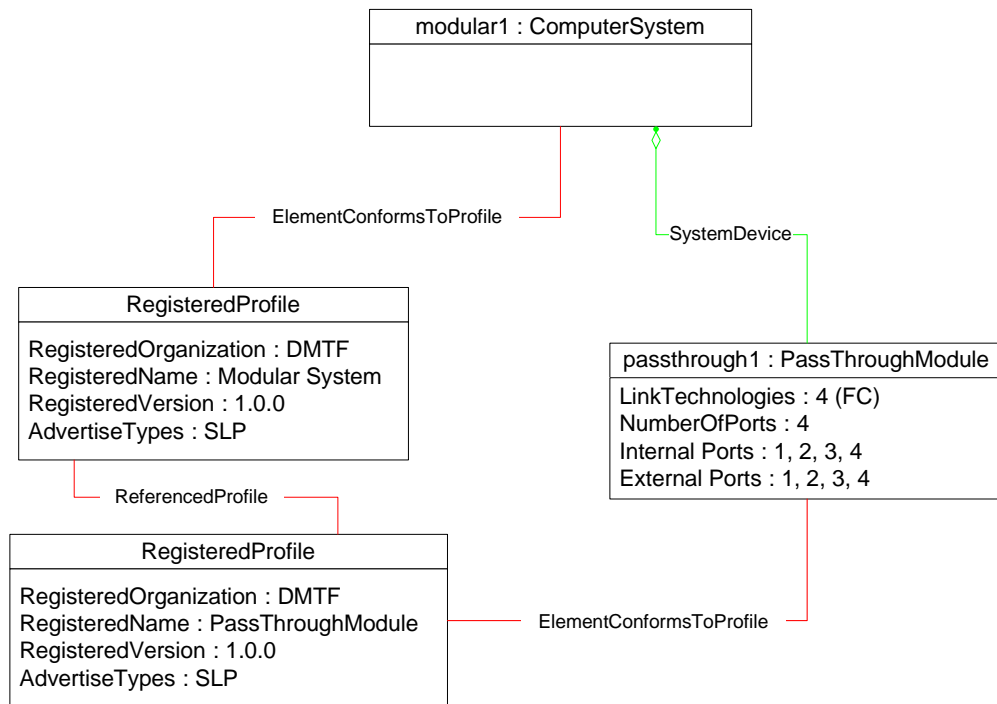
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

366 **9 Use Cases**

367 This section outlines the use cases specific to pass-through modules. Use cases for functionality that is
 368 not specific to modular systems are documented in the profiles for that functionality. Use cases are
 369 informative and not intended to define the requirements for conformance.

370 **9.1 Object Diagrams**

371 The object diagram in Figure 2 illustrates a single pass-through module installed in a modular system.
 372 *passthrough1* is conformant with the *Pass-Through Module Profile* as indicated by the
 373 *CIM_ElementConformsToProfile* association that references the instance. This instance of
 374 *CIM_PassThroughModule* represents a four-port, fiber-channel pass-through module. The internal and
 375 external ports are directly mapped, as indicated by the corresponding array positions for each internal
 376 and external port index.

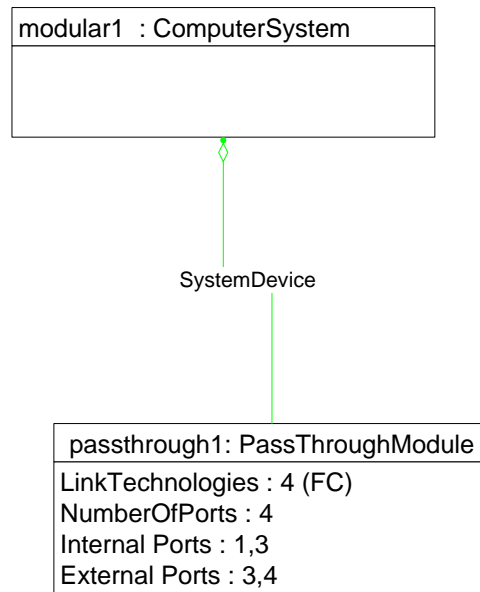


377

378

Figure 2 – Instance Diagram

379 The object diagram in Figure 3 illustrates a single pass-through module installed in a modular system.
 380 *passthrough1* is a four-port, fiber-channel pass-through module. Internal port 1 is mapped to external port
 381 3. Internal port 3 is mapped to external port 4. Internal ports 2 and 4 and external ports 1 and 2 are not
 382 mapped.



383

384

Figure 3 – Port Mappings Crossed

385 9.2 Determine Pass-Through Module Link Technology

386 A client can determine the link technology or technologies supported by a pass-through module by
 387 querying the value of the CIM_PassThroughModule.LinkTechnologies property.

388 9.3 Determine Pass-Through Module Port Mappings

389 A client can determine which internal ports are mapped to which external ports as follows:

- 390 1) Query the value of the CIM_PassThroughModule.InternalPorts property.
- 391 2) Query the value of the CIM_PassThroughModule.ExternalPorts property.
- 392 3) Compare the corresponding array indices of each property.

393 The port number at array index *x* of the InternalPorts property will be mapped to the port number at array
 394 index *x* of the ExternalPorts property.

395 9.4 Determine Whether Port Mappings Are Configurable

396 A client can determine whether port mappings are configurable by querying the value of the
 397 IsProgrammable property of the CIM_PassThroughModule instance. A value of TRUE indicates that the
 398 ports are configurable and the AssignPorts() method will be supported.

399 9.5 Manage Port Mappings on a Pass-Through Module

400 A client can manage the port mappings on an instance of CIM_PassThroughModule as follows:

- 401 1) Verify that port mappings are configurable as described in section 9.4.
- 402 2) If port mappings are configurable, invoke the AssignPorts() method with the target internal port,
 403 external port, and a flag that indicates whether the ports should be mapped or unmapped from
 404 each other.

405 9.6 Determining If ElementName Can Be Modified

406 For a given instance of CIM_PassThroughModule, a client can determine whether it can modify the
407 ElementName as follows:

- 408 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target
409 instance.
- 410 2) Query the value of the ElementNameEditSupported property of the
411 CIM_EnabledLogicalElementCapabilities instance. If the value is TRUE, the client can modify
412 the ElementName property of the target instance.

413 9.7 Determining If State Management Is Supported

414 For a given instance of CIM_PassThroughModule, a client can determine whether state management is
415 supported as follows:

- 416 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target
417 instance.
- 418 2) Query the value of the RequestedStatesSupported property. If at least one value is specified,
419 state management is supported.

420 10 CIM Elements

421 Table 9 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
422 implemented as described in Table 9. Sections 7 (“Implementation Requirements”) and 8 (“Methods”)
423 may impose additional requirements on these elements.

424 **Table 9 – Required CIM Elements: Pass-Through Module Profile**

Element Name	Requirement	Notes
Classes		
CIM_ElementCapabilities	Conditional	See section 10.1.
CIM_EnabledLogicalElementCapabilities	Optional	See section 10.2.
CIM_PassThroughModule	Mandatory	See section 10.3.
CIM_RegisteredProfile	Mandatory	See section 10.4.
CIM_SystemDevice	Mandatory	See section 0.
Indications		
None defined in this profile		

425 **10.1 CIM_ElementCapabilities**

426 CIM_ElementCapabilities is used to associate CIM_PassThroughModule with the instance of
 427 CIM_EnabledLogicalElementCapabilities that describes the capabilities of the pass-through module.
 428 CIM_ElementCapabilities is conditional on the instantiation of CIM_EnabledLogicalElementCapabilities.

429 Table 10 provides information about the properties of CIM_ElementCapabilities.

430 **Table 10 – Class: CIM_ElementCapabilities**

Properties	Requirement	Notes
ManagedElement	Mandatory	Shall reference the Central Instance Cardinality 1..*
Capabilities	Mandatory	Shall reference the instance of CIM_EnabledLogicalElementCapabilities Cardinality 0..1

431 **10.2 CIM_EnabledLogicalElementCapabilities**

432 CIM_EnabledLogicalElementCapabilities represents the capabilities of the pass-through module.
 433 Table 11 provides information about the properties of CIM_EnabledLogicalElementCapabilities.

434 **Table 11 – Class: CIM_EnabledLogicalElementCapabilities**

Properties	Requirement	Notes
InstanceID	Mandatory	None
RequestedStatesSupported	Mandatory	See sections 7.1.1.1.1 and 7.1.2.1.1.
ElementNameEditSupported	Mandatory	See sections 7.1.3.1.1 and 7.1.4.1.1.
MaxElementNameLen	Conditional	See sections 7.1.3.1.2 and 7.1.4.1.2.

435 **10.3 CIM_PassThroughModule**

436 CIM_PassThroughModule represents a pass-through module. Table 12 provides information about the
 437 properties of CIM_PassThroughModule.

438 **Table 12 – Class: CIM_PassThroughModule**

Properties and Methods	Requirement	Notes
LinkTechnologies	Mandatory	None
OtherLinkTechnologies	Conditional	When an array index of LinkTechnologies contains the value "Other", the same array index of this property shall contain a value that identifies the link technology.
Is Programmable	Mandatory	None
NumberOfPorts	Mandatory	None
InternalPorts	Mandatory	None
ExternalPorts	Mandatory	None
SystemCreationClassName	Mandatory	None
SystemName	Mandatory	None
CreationClassName	Mandatory	None
DeviceID	Mandatory	None
EnabledState	Mandatory	See sections 7.1.1.3 and 7.1.2.3.
RequestedState	Mandatory	See sections 7.1.1.2 and 7.1.2.2.
RequestStateChange()	Conditional	See section 8.2.
AssignPorts()	Conditional	See sections 7.2 and 8.1.

439 **10.4 CIM_RegisteredProfile**

440 CIM_RegisteredProfile identifies the *Pass-Through Module Profile* in order for a client to determine
 441 whether an instance of CIM_ComputerSystem is conformant with this profile. The CIM_RegisteredProfile
 442 class is defined by the [Profile Registration Profile](#). With the exception of the mandatory values specified
 443 for the properties in Table 13, the behavior of the CIM_RegisteredProfile instance is in accordance with
 444 the constraints specified in the [Profile Registration Profile](#).

445 **Table 13 – Class: CIM_RegisteredProfile**

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Pass-Through Module".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

446 NOTE: Previous versions of this document included the suffix "Profile" for the RegisteredName value. If
 447 implementations querying for the RegisteredName value find the suffix "Profile", they should ignore the suffix, with
 448 any surrounding white spaces, before any comparison is done with the value as specified in this document.

449

450 **10.5 CIM_SystemDevice**

451 CIM_SystemDevice is used to associate an instance of CIM_PassThroughModule with an instance of
 452 CIM_ComputerSystem that represents a modular enclosure. Table 14 provides information about the
 453 properties of CIM_SystemDevice.

454 **Table 14 – Class: CIM_SystemDevice**

Properties	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to the Scoping Instance. Cardinality 1
PartComponent	Mandatory	This property shall be a reference to the Central Instance. Cardinality 1..*

455

456
457
458
459

ANNEX A
(informative)
Change Log

Version	Date	Description
1.0.0	6/16/2009	DMTF Standard Release

460
461