



1

2

3

4

5

6

7

8

9 Copyright Notice

10 Copyright © 2007, 2005 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

11 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems  
12 management and interoperability. Members and non-members may reproduce DMTF specifications and  
13 documents for uses consistent with this purpose, provided that correct attribution is given. As DMTF  
14 specifications may be revised from time to time, the particular version and release date should always be  
15 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

## CONTENTS

31	Foreword .....	6
32	Introduction .....	7
33	1 Scope .....	9
34	2 Normative References .....	9
35	2.1 Approved References .....	9
36	2.2 References under Development .....	9
37	2.3 Other References .....	9
38	3 Terms and Definitions .....	9
39	4 Symbols and Abbreviated Terms .....	11
40	5 Synopsis .....	11
41	6 Description .....	11
42	6.1 Whitebox Policies .....	13
43	6.2 Blackbox Policies .....	14
44	7 Implementation .....	14
45	7.1 CIM_QueryCondition (Optional) .....	15
46	7.2 CIM_MethodAction (Optional) .....	15
47	7.3 CIM_CompoundPolicyAction (Optional) .....	15
48	7.4 CIM_CompoundPolicyCondition (Optional) .....	15
49	7.5 CIM_ElementInPolicyRoleCollection (Optional) .....	16
50	7.6 CIM_PolicyAction (Optional) .....	16
51	7.7 CIM_PolicyActionInPolicyAction (Optional) .....	16
52	7.8 CIM_PolicyActionInPolicyRule (Optional) .....	17
53	7.9 CIM_PolicyCondition (Optional) .....	17
54	7.10 CIM_PolicyConditionInPolicyCondition (Optional) .....	17
55	7.11 CIM_PolicyConditionInPolicyRule (Optional) .....	17
56	7.12 CIM_PolicyRoleCollection (Optional) .....	18
57	7.13 CIM_PolicyRoleCollectionInSystem (Conditional) .....	18
58	7.14 CIM_PolicyRule .....	18
59	7.15 CIM_PolicyRuleInSystem .....	19
60	7.16 CIM_PolicySetAppliesToElement (Optional) .....	19
61	7.17 CIM_PolicySetComponent (Optional) .....	19
62	7.18 CIM_PolicySetInRoleCollection (Optional) .....	19
63	7.19 CIM_PolicySetValidityPeriod (Optional) .....	19
64	7.20 CIM_PolicyTimePeriodCondition (Optional) .....	20
65	7.21 CIM_ReusablePolicy (Optional) .....	22
66	7.22 CIM_ReusablePolicyContainer (Optional) .....	22
67	8 Methods .....	22
68	8.1 CIM_PolicyRoleCollection. ActivatePolicySet() .....	22
69	8.2 CIM_PolicyRoleCollection. DeactivatePolicySet() .....	22
70	8.3 Profile Conventions for Operations .....	23
71	8.4 CIM_QueryCondition .....	24
72	8.5 CIM_MethodAction .....	24
73	8.6 CIM_PolicyRule .....	24
74	8.7 CIM_PolicyCondition .....	24
75	8.8 CIM_PolicyAction .....	24
76	8.9 CIM_PolicySetAppliesToElement .....	24
77	8.10 CIM_PolicyConditionInPolicyRule .....	24
78	8.11 CIM_PolicyActionInPolicyRule .....	25
79	8.12 CIM_CompoundPolicyAction .....	25
80	8.13 CIM_CompoundPolicyCondition .....	25
81	8.14 CIM_ElementInPolicyRoleCollection .....	25
82	8.15 CIM_PolicyActionInPolicyAction .....	25

83	8.16	CIM_PolicyConditionInPolicyCondition .....	26
84	8.17	CIM_PolicyRoleCollection .....	26
85	8.18	CIM_PolicyRoleCollectionInSystem .....	26
86	8.19	CIM_PolicyRuleInSystem .....	27
87	8.20	CIM_PolicySetComponent .....	27
88	8.21	CIM_PolicySetInRoleCollection .....	27
89	8.22	CIM_PolicySetValidityPeriod .....	28
90	8.23	CIM_PolicyTimePeriodCondition .....	28
91	8.24	CIM_ReusablePolicy .....	28
92	8.25	CIM_ReusablePolicyContainer .....	28
93	9	Use Cases .....	28
94	9.1	Object Diagrams .....	28
95	9.2	Blackbox Policy Object Diagram .....	29
96	9.3	Propagate HealthState to OperationalStatus .....	30
97	10	CIM Elements .....	30
98	10.1	CIM_QueryCondition .....	32
99	10.2	CIM_MethodAction .....	32
100	10.3	CIM_PolicyRule .....	33
101	10.4	CIM_PolicyCondition .....	33
102	10.5	CIM_PolicyAction .....	34
103	10.6	CIM_PolicySetAppliesToElement .....	34
104	10.7	CIM_PolicyConditionInPolicyRule .....	34
105	10.8	CIM_PolicyActionInPolicyRule .....	35
106	10.9	CIM_CompoundPolicyAction .....	35
107	10.10	CIM_CompoundPolicyCondition .....	36
108	10.11	CIM_ElementInPolicyRoleCollection .....	36
109	10.12	CIM_PolicyActionInPolicyAction .....	37
110	10.13	CIM_PolicyConditionInPolicyCondition .....	37
111	10.14	CIM_PolicyRoleCollection .....	37
112	10.15	CIM_PolicyRoleCollectionInSystem .....	38
113	10.16	CIM_PolicyRuleInSystem .....	38
114	10.17	CIM_PolicySetComponent .....	38
115	10.18	CIM_PolicySetInRoleCollection .....	39
116	10.19	CIM_PolicySetValidityPeriod .....	39
117	10.20	CIM_PolicyTimePeriodCondition .....	40
118	10.21	CIM_ReusablePolicy .....	40
119	10.22	CIM_ReusablePolicyContainer .....	41
120	ANNEX A (Informative)	Change Log .....	42
121	ANNEX B (informative)	Acknowledgments .....	43
122			
123			
124	Figure 1 – Policy Profile:	Class Diagram .....	13
125	Figure 2 – Blackbox Health	Propagation Policy Diagram .....	29
126	Figure 3 – Health Propagation	Policy Object Diagram .....	30
127			
128			
129	Table 1 – Referenced Profiles .....		11
130	Table 2 – Operations: CIM_PolicySetAppliesToElement .....		24
131	Table 3 – Operations: CIM_PolicyConditionInPolicyRule .....		24

132	Table 4 – Operations: CIM_PolicyActionInPolicyRule.....	25
133	Table 5 – Operations: CIM_ElementInPolicyRoleCollection.....	25
134	Table 6 – Operations: CIM_PolicyActionInPolicyAction.....	25
135	Table 7 – Operations: CIM_PolicyConditionInPolicyCondition.....	26
136	Table 8 – Operations: CIM_PolicyRoleCollectionInSystem.....	26
137	Table 9 – Operations: CIM_PolicyRuleInSystem.....	27
138	Table 10 – Operations: CIM_PolicySetComponent.....	27
139	Table 11 – Operations: CIM_PolicySetInRoleCollection.....	27
140	Table 12 – Operations: CIM_PolicySetValidityPeriod.....	28
141	Table 13 – Operations: CIM_ReusablePolicy.....	28
142	Table 14 – CIM Elements: Policy Profile.....	30
143	Table 15 – Class: CIM_QueryCondition.....	32
144	Table 16 – Class: CIM_MethodAction.....	32
145	Table 17 – Class: CIM_PolicyRule.....	33
146	Table 18 – Class: CIM_PolicyCondition.....	33
147	Table 19 – Class: CIM_PolicyAction.....	34
148	Table 20 – Class: CIM_PolicySetAppliesToElement.....	34
149	Table 21 – Class: CIM_PolicyConditionInPolicyRule.....	35
150	Table 22 – Class: CIM_PolicyActionInPolicyRule.....	35
151	Table 23 – Class: CIM_CompoundPolicyAction.....	36
152	Table 24 – Class: CIM_CompoundPolicyCondition.....	36
153	Table 25 – Class: CIM_ElementInPolicyRoleCollection.....	37
154	Table 26 – Class: CIM_PolicyActionInPolicyAction.....	37
155	Table 27 – Class: CIM_PolicyConditionInPolicyCondition.....	37
156	Table 28 – Class: CIM_PolicyRoleCollection.....	38
157	Table 29 – Class: CIM_PolicyRoleCollectionInSystem.....	38
158	Table 30 – Class: CIM_PolicyRuleInSystem.....	38
159	Table 31 – Class: CIM_PolicySetComponent.....	39
160	Table 32 – Class: CIM_PolicySetInRoleCollection.....	39
161	Table 33 – Class: CIM_PolicySetValidityPeriod.....	40
162	Table 34 – Class: CIM_PolicyTimePeriodCondition.....	40
163	Table 35 – Class: CIM_ReusablePolicy.....	41
164	Table 36 – Class: CIM_ReusablePolicyContainer.....	41

165

## Foreword

167 The *Policy Profile* (DSP1003) was prepared by the Policy Working Group of the DMTF.

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

## Introduction

171 The information in this specification should be sufficient for a provider or consumer of this data to  
172 unambiguously identify the classes, properties, methods, and values that shall be instantiated and  
173 manipulated to represent and manage policy collections and rules modeled using the DMTF Common  
174 Information Model (CIM) core and extended model definitions.

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





177

178

179 The *Policy Profile* is an abstract profile that extends the management capability of referencing profiles by  
180 adding the capability to represent policies that apply to Managed Elements and optionally define specific  
181 CQL or CIM-SPL statements that represent the policies. The *Policy Profile* is to be specialized by profiles  
182 that define specific policy algorithms and optionally define CQL or CIM-SPL statements for particular  
183 policies.

184

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

188

189 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*

190 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*

191 DMTF [DSP1000](#), *Management Profile Specification Template*

192 DMTF [DSP1001](#), *Management Profile Specification Usage Guide*

193

194

195 DMTF [DSP1033](#), *Profile Registration Profile*

196 DMTF [DSP0202](#), *CIM Query Language Specification*

197 DMTF [DSP0231](#), *CIM Simplified Policy Language*

198

199

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

202 Unified Modeling Language (UML) from the Open Management Group (OMG), <http://www.uml.org>

203

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

206

207

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

209  
210  
211 used for statements of possibility and capability, whether material, physical, or causal  
212  
213  
214 indicates requirements to be followed strictly to conform to the document when the specified conditions  
215 are met  
216  
217  
218 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
219 permitted  
220  
221  
222 indicates a course of action permissible within the limits of the document  
223  
224  
225 indicates a course of action permissible within the limits of the document  
226  
227  
228 indicates a course of action permissible within the limits of the document  
229  
230  
231 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
232 "Referenced Profiles" table  
233  
234  
235 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
236 permitted  
237  
238  
239 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
240 permitted  
241  
242  
243 indicates that among several possibilities, one is recommended as particularly suitable, without  
244 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required  
245  
246  
247 indicates that a certain possibility or course of action is deprecated but not prohibited  
248  
249  
250 indicates that this profile does not define any constraints for the referenced CIM element or operation

251  
252  
253 refers to CQL and CIM-SPL as the two defined languages

254

255

256  
257  
258 CIM Query Language

259  
260  
261 CIM Simplified Policy Language

262

263 Profile Name: *Policy*

264 Version: 1.0.0e

265 Organization: DMTF

266 CIM schema version: 2.12

267 Central Class: CIM\_PolicyRule

268 Scoping Class: CIM\_System

269 This abstract profile specification shall not be directly implemented; implementations shall be based on a  
270 profile specification that specializes the requirements of this profile.

271 The *Policy Profile* is an abstract profile that extends the management capability of referencing profiles by  
272 adding the capability to represent policies that apply to Managed Elements and optionally define specific  
273 CQL or CIM-SPL statements that represent the policies. The profile is to be specialized by profiles that  
274 define specific policy algorithms and optionally define CQL or CIM-SPL statements for particular policies.

275 Table 1 lists profiles upon which this profile has a dependency.

276



277

278 The *Policy Profile* is an abstract profile that defines the CIM elements and constraints for specifying  
279 policies that affect Managed Elements. Architecture/implementation details are out of the scope of this  
280 abstract profile.

281 A *policy* defines actions that are taken under a set of conditions. Generally, a policy rule is automated. A  
282 policy defines what is to be done in the users stead and under what circumstances. The policy rule itself  
283 is generally not the entity that carries out the action. Instead, a policy rule is assumed to be the entity that  
284 directs the change. Just as when an end user requests that a host reboot, that user is not the entity that

285 informs the host hardware to reboot directly; policy rules request that the action be taken on its behalf. As  
286 such, a policy rule reacts to its own environment.

287 A condition is expressed as the state of one or more Managed Elements. The description of the state  
288 could be simple or complex. It can be described in words or could be defined in terms of values of the  
289 properties of the CIM instances.. Regardless of the complexity of the state in the condition, the condition  
290 itself defines state criteria that are considered as a whole. If the states of the Managed Elements are as  
291 defined in the condition, then the condition itself is evaluated as TRUE. Conditions could be notified of a  
292 state change that is triggered.

293 An *action* defines how the state ought to change as a result of the action itself being invoked. A policy  
294 action defines that a single action is taken. Actions can succeed or fail. The action could be described in  
295 words or defined in terms of the properties or methods of CIM instances.

296 A policy rule is evaluated. A policy rule can be assumed to be executed periodically. A PolicyRule may be  
297 associated with one or more policy time periods, indicating the schedule according to which the policy  
298 rule is active. At all times that fall outside these time periods, the PolicyRule is inactive. A PolicyRule is  
299 treated as active at ALL times, if it does not specify a PolicyTimePeriodCondition. The policy rule is  
300 executed by a special type of policy that depends on time or could be executed on a schedule defined  
301 elsewhere. Otherwise, a policy rule could be evaluated as a result of its trigger conditions being notified of  
302 a state change.

303 During the policy evaluation, the states defined in the associated conditions are compared to the actual  
304 states of the related Managed Elements. Each condition is considered in order. The order could be  
305 defined explicitly or implicitly. Implicitly ordered conditions are those that depend on the output of the  
306 evaluation of other conditions in order to be evaluated. If enough conditions are evaluated to be TRUE,  
307 the actions are then taken. Likewise, actions could be ordered explicitly or implicitly. Implicitly ordered  
308 actions are those that require the evaluation of conditions or execution of actions for themselves to be  
309 executed.

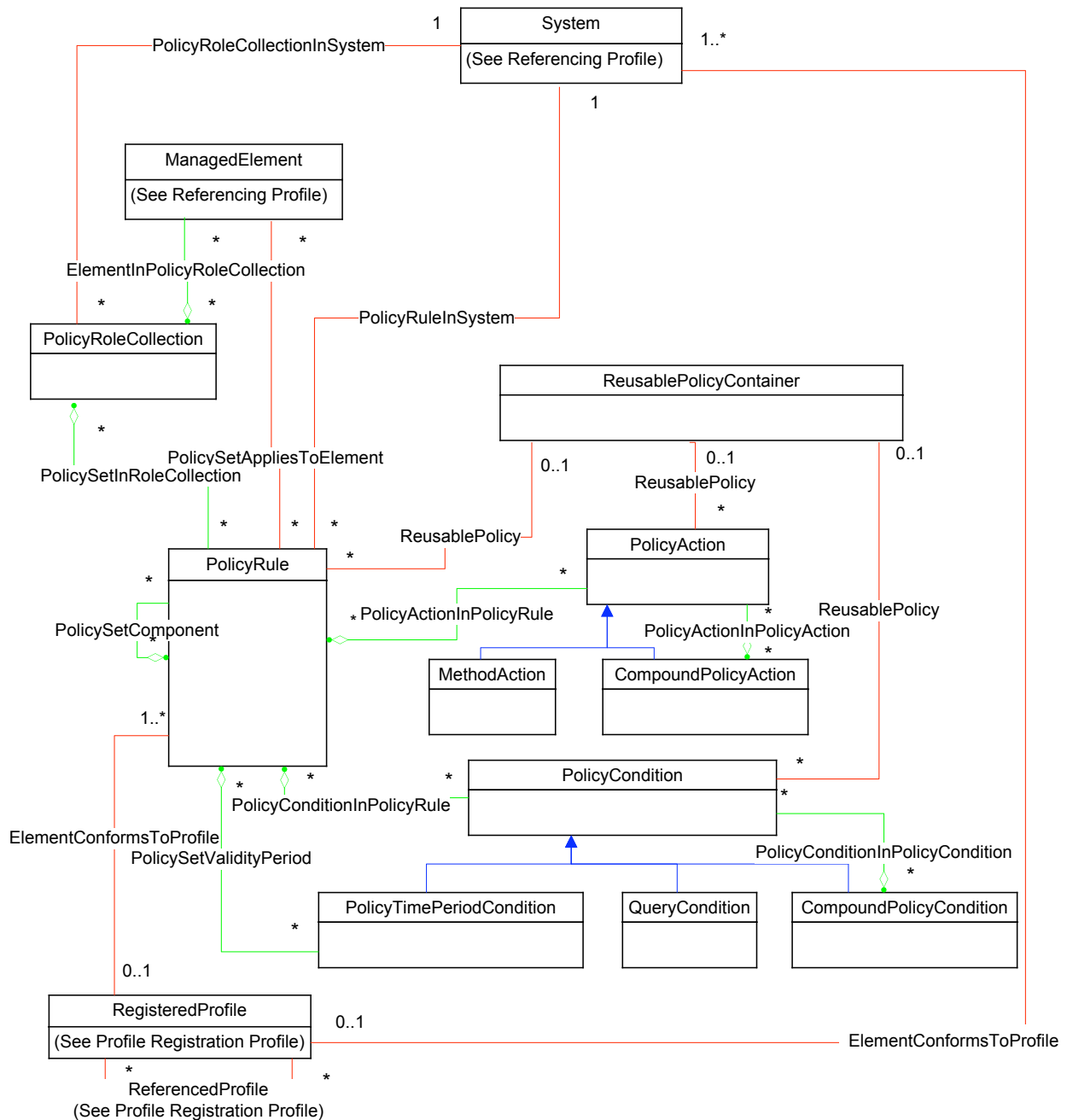
310 Policy rules must be defined in terms of a response. The policy rule must be defined without depending  
311 on other policy rules.

312 Figure 1 represents the class schema for the *Policy Profile*. For simplicity, the prefix CIM\_ has been  
313 removed from the names of the classes.

314 Functionality within the scope of this profile includes:

- 315 • a specification of the CIM\_PolicyRule class and optional related CIM\_PolicyCondition and  
316 CIM\_PolicyAction classes

317 Functionality explicitly excluded from the scope of this profile includes how the specific CQL or CIM-SPL  
318 definitions for policy conditions and actions might be modified by the client. This is defined further in  
319 specialized profile.



320

321

322

323 The term “Whitebox” is applied to policies that have policy rules that can be inspected. Whitebox policy  
 324 rules are defined in terms of policy language statements that directly interact with the model. As such,  
 325 special subclasses of CIM\_PolicyCondition and CIM\_PolicyAction, CIM\_QueryCondition and  
 326 CIM\_MethodAction respectively, are used to contain the expression of how the CIM representation or  
 327 state of the model causes the policy to act, and what CIM action is imposed on the model as a result.

328 The benefit of Whitebox policies is that they can be inspected, they are programmatically deterministic,  
 329 and they have precisely the same meaning and result to observers without any prior knowledge. Given

330 the CIM and the CQL standard and this definition of the behavior and rules of policies, the policy rule  
331 algorithms as expressed in CQL can be interpreted by both the CIM clients that read them and the  
332 systems that host and evaluate them.

333 A CIM\_QueryCondition instance contains a CQL statement that itself expresses the state whose  
334 attainment results in the condition being evaluated as TRUE. When such a CQL statement is evaluated,  
335 the CQL is executed and a result is produced. This result is available to other conditions and actions that  
336 are members of the policy rule. If the Query Condition produces an empty result, it is considered to be  
337 evaluated as FALSE. The result itself can be referenced as a CIM class in CQL construct by using its  
338 QueryResultName property.

339 A CIM\_QueryCondition may be dependent on the results of other queries. Specifically, a condition may  
340 be dependent on other conditions. This result may be joined with other classes or other results to produce  
341 additional results. A dependence of several conditions might be produced such that the set resembles a  
342 decision tree. The least dependent conditions must be evaluated first. The most dependent conditions  
343 must be evaluated last.

344 A CIM\_MethodAction instance defines the query that itself builds an intrinsic or extrinsic method call. The  
345 components of this query must contain first the object name of the method to be called and, thereafter, a  
346 list of named parameters. The parameter names and types must match those of the method declaration.  
347 Type conversion is supplied by the associated CQL implementation. The result of the selected list applies  
348 to the referenced method as input parameters. If the result of the query produces an empty result, the  
349 method will not be executed. Such an execution must be considered successful. The method is executed  
350 once for every row of the results, which in turn provides the parameters for each method call.

351 When the CIM\_MethodAction query executes, it must apply its action to the model. The result of the  
352 method execution shall be the list of output parameters for the method; each, in turn, shall be identifiable  
353 by its parameter name.

354 The query result of the conditions and the method result of the action may be used by other actions to  
355 construct method calls.

356 Such policies can take several execution paths based on the conditions that are evaluated as TRUE and  
357 the actions taken as a result. The effect of some of the conditions evaluated as TRUE may result in only  
358 some of the actions being taken. For a given policy rule, a different set of conditions may produce a  
359 different set of actions depending on the evaluation.

360

361 The “Blackbox” term is applied to policies that have condition and action algorithms defined but cannot be  
362 inspected through the CIM model. Blackbox policy rules are represented in CIM by policy rule names or  
363 labels, but they are not accompanied by classes that contain specific condition or action query strings.  
364 The model manipulation algorithms that represent a Blackbox policy is defined in a profile that describes  
365 the algorithms in sufficient detail for consistent implementation by instrumentation.

366 Determining whether a policy rule represents a blackbox policy can be accomplished by inspecting  
367 whether an instance of CIM\_RegisteredProfile, which specifies a blackbox policy is associated with an  
368 instance of CIM\_PolicyRule, which represents the policy rule in question, via the  
369 CIM\_ElementConformsToProfile association.

370

371 This section details requirements and guidelines for propagating and formulating certain properties of the  
372 classes of this profile. Required methods are described in section 8 (“Methods”), and properties are  
373 described in section 10 (“CIM Elements”).

374

375 Instances of CIM\_ QueryCondition are optional and may be used to define the criteria for generating a set  
376 of query results that are accessible to other QueryConditions or MethodActions of the same PolicyRule.

377

378

379 Instances of CIM\_ MethodAction are optional and may be used to invoke methods as defined by a query.

380

381

382 Instances of CIM\_ CompoundPolicyAction are optional and may be used to represent an expression that  
383 consists of an ordered sequence of actions represented by instances of the CIM\_ PolicyAction class.

384 The ordering of action terms shall be represented by instances of the CIM\_ PolicyActionInPolicyAction  
385 aggregation.

386

387 Each action in an explicit action sequence is represented by a CIM\_ PolicyAction instance aggregated by  
388 a CIM\_ PolicyActionInPolicyAction instance.

389 The order of the actions in the sequence shall be specified by the ActionOrder property of each  
390 CIM\_ PolicyActionInPolicyAction aggregation instance. An ActionOrder property value of 0 shall mean that  
391 the aggregated CIM\_ PolicyAction instance may be executed at any time in the execution sequence. Non-  
392 zero values of ActionOrder indicate a relative priority, in which lower numbers shall be executed before  
393 higher numbers. Two or more instances of CIM\_ PolicyAction aggregated with the same non-zero value  
394 may be executed in any order, but after all instances associated with lower non-zero values and before all  
395 instances with higher non-zero values.

396 The optional SequencedActions property of CIM\_ CompoundPolicyAction specifies the order of execution  
397 of the actions in the sequence.

398 The optional ExecutionStrategy property of CIM\_ CompoundPolicyAction defines the strategy to be used  
399 in executing the action sequence.

400

401

402 Instances of CIM\_ CompoundPolicyCondition are optional and shall be used to represent a condition  
403 expression. Instances of CIM\_ PolicyCondition are composed into a condition expression by instances of  
404 the CIM\_ PolicyConditionInPolicyCondition aggregation.

405 The CIM\_ PolicyCondition instances shall be aggregated into an instance of  
406 CIM\_ CompoundPolicyCondition forming a set of conditions.

407 ConditionListType is an optional property of a CIM\_ CompoundPolicyCondition instance. It is used with  
408 the GroupNumber and ConditionNegated properties of each CIM\_ PolicyConditionInPolicyCondition  
409 aggregation instance to collectively specify an explicit condition expression. The set of individual terms of  
410 an explicit condition expression is represented by the aggregated CIM\_ PolicyCondition instances.

411 The set of terms is divided into one or more subsets. Two instances of CIM\_ PolicyCondition that belong  
412 to the same explicit condition expression shall have the same value for the  
413 CIM\_ PolicyConditionInPolicyCondition.GroupNumber property for the instances of

414 CIM\_PolicyConditionInPolicyCondition that associate them to an instance of  
415 CIM\_CompoundPolicyCondition.

416 The ConditionNegated property specifies whether the associated condition must be negated when  
417 evaluated in the condition expression.

418 The ConditionListType property shall have one of the following two values:

- 419 • Disjunctive Normal Form (DNF): Each term is evaluated for TRUE or FALSE. If the  
420 ConditionNegated property is TRUE for the term, the term's evaluation result is negated. Then,  
421 the evaluation result for each term that belongs to a particular subset (the same GroupNumber)  
422 is joined together with AND. Finally the result of evaluating each subgroup is joined together  
423 with OR.
- 424 • Conjunctive Normal Form (CNF): Each term is evaluated for TRUE or FALSE. If the  
425 ConditionNegated property is TRUE for the term, the term's evaluation result is negated. Then,  
426 the evaluation result for each term that belongs to a particular subset (the same GroupNumber)  
427 is joined together with OR. Finally the result of evaluating each subgroup is joined together with  
428 AND.

429 The Disjunctive Normal Form (DNF) is the default value for ConditionListType.

430

431 An instance of CIM\_ElementInPolicyRoleCollection is an optional aggregation that may be used to  
432 associate a CIM\_ManagedElement instance with an instance of CIM\_PolicyRoleCollection.

433 Use of this association shall indicate that instances of CIM\_PolicySet, which are aggregated through  
434 CIM\_PolicySetInRoleCollection into the referenced instance of CIM\_PolicyRoleCollection, may be applied  
435 to the referenced instance of CIM\_ManagedElement.

436 Constraints on the behavior of the policy as applied to instances of CIM\_ManagedElement may be further  
437 specified in a specialized policy profile.

438

439

440 Instances of the abstract class CIM\_PolicyAction are optional. Each instance represents an explicit  
441 specification of an action.

442 CIM\_PolicyAction instances may be aggregated by CIM\_PolicyActionInPolicyRule into one or more  
443 instances of CIM\_PolicyRule or by CIM\_PolicyActionInPolicyAction into one or more instances of  
444 CIM\_CompoundPolicyAction. These associations, together with the aggregating instances, define the  
445 execution order of an aggregated set of CIM\_PolicyAction instances.

446 If a CIM\_PolicyAction instance may be reused, it shall be aggregated into one  
447 CIM\_ReusablePolicyContainer instance.

448

449 Depending on rules within the aggregating CIM\_CompoundPolicyAction instance, the execution  
450 sequence of the set of aggregated CIM\_PolicyAction instances may be enforced. In that case, the value  
451 of the optional ActionOrder property of each instance affects the execution sequence of the aggregated  
452 CIM\_PolicyAction instance. If the ActionOrder property is not specified, or if it is specified with a value of  
453 NULL, the property shall have the value 0.

454 If the order of action is enforced, the execution sequence covers the set of CIM\_PolicyAction instances  
455 that are associated through CIM\_PolicyActionInPolicyAction to the same instance of  
456 CIM\_CompoundPolicyAction. In this case, a value of 0 shall mean that the aggregated CIM\_PolicyAction



457 instances may be executed at any time in the execution sequence. Non-zero values of ActionOrder  
458 indicate a relative priority, in which lower numbers shall be executed before higher numbers. Two or more  
459 instances of CIM\_PolicyAction aggregated with the same non-zero value may be executed in any order,  
460 but after all instances associated with lower non-zero values and before all instances with higher non-  
461 zero values.

462

463 An instance of CIM\_PolicyActionInPolicyRule is an optional aggregation that may be used to explicitly  
464 specify that an action, represented by an instance of CIM\_PolicyAction, shall be executed if the  
465 conditions specified for the referenced instance of CIM\_PolicyRule are met.

466 Depending on rules within the aggregating CIM\_PolicyRule instance, the execution sequence of the set of  
467 aggregated CIM\_PolicyAction instances may be enforced. In that case, the value of the optional  
468 ActionOrder property of each instance affects the execution sequence of the aggregated  
469 CIM\_PolicyAction instances. If the ActionOrder property is not specified, or if it is specified with a value of  
470 NULL, the property shall have the value 0.

471 If the order of action is enforced, the execution sequence covers the set of CIM\_PolicyAction instances  
472 that are associated by CIM\_PolicyActionInPolicyRule to the same instance of CIM\_PolicyRule. In this  
473 case, a value of 0 shall mean that the aggregated CIM\_PolicyAction instances may be executed at any  
474 time in the execution sequence. Non-zero values of ActionOrder indicate a relative priority, in which lower  
475 numbers shall be executed before higher numbers. Two or more instances of CIM\_PolicyAction  
476 aggregated with the same non-zero value may be executed in any order, but after all instances  
477 associated with lower non-zero values and before all instances with higher non-zero values.

478

479 Instances of the abstract class CIM\_PolicyCondition are optional. Each instance represents an explicit  
480 specification of a condition.

481 CIM\_PolicyCondition instances may be aggregated by CIM\_PolicyConditionInPolicyRule into one or more  
482 instances of CIM\_PolicyRule or by CIM\_PolicyConditionInPolicyCondition into one or more instances of  
483 CIM\_CompoundPolicyCondition. These associations, together with the aggregating instances, define an  
484 expression made up of a set of condition terms represented by CIM\_PolicyCondition instances.

485 If a CIM\_PolicyCondition instance may be reused, it shall be aggregated into one  
486 CIM\_ReusablePolicyContainer instance.

487

488 The GroupNumber is an optional property used to produce subsets of the set of condition terms. If the  
489 GroupNumber property is not specified, or if it is specified with a value of NULL, the property shall have  
490 the value 0.

491 ConditionNegated is an optional property used to indicate whether the aggregated CIM\_PolicyCondition  
492 instance should be negated (TRUE) or not (FALSE). If the ConditionNegated property is not specified, or  
493 if it is specified with a value of NULL, the property shall have the value FALSE.

494

495 The CIM\_PolicyCondition instances aggregated into an instance of CIM\_PolicyRule form a set of  
496 condition terms.

497 GroupNumber is an optional property that may be used to produce subsets of the set of condition terms.  
498 If the GroupNumber property is not specified, or if it is specified with a value of NULL, the property shall  
499 have the value 0.

500 ConditionNegated is an optional property used to indicate whether the aggregated CIM\_PolicyCondition  
501 instance shall be negated (TRUE) or not (FALSE). If the ConditionNegated property is not specified, or if  
502 it is specified with a value of NULL, the property shall have the value FALSE.

503

504 Instances of CIM\_PolicyRoleCollection are optional collections of CIM\_PolicyRule instances that  
505 collectively meet a particular set of policy roles within a system.

506 These policy roles shall be represented using the Role Property. The value shall have the form:  
507 <RoleName>["&&"<RoleName>]\*, where the individual role names appear in alphabetical order  
508 (according to the collating sequence for UCS-2). Implementations may treat Role property values that are  
509 specified as "role combinations" as simple strings.

510 Each instance of CIM\_PolicyRoleCollection shall be associated with exactly one CIM\_System instance  
511 through an instance of CIM\_PolicyRoleCollectionInSystem.

512 Each CIM\_PolicyRule instance required to meet a particular policy role shall be aggregated into an  
513 instance of CIM\_PolicyRoleCollection by an instance of CIM\_PolicySetInRoleCollection.

514

515 An instance of CIM\_PolicyRoleCollectionInSystem is a conditional association that shall be used to  
516 establish a relationship between an instance of CIM\_PolicyRoleCollection and a Scoping Instance, such  
517 as an AdminDomain or ComputerSystem. When the optional behavior specified in section 7.13 is  
518 supported, the instance of this class is used to associate the in an instance of CIM\_PolicyRoleCollection  
519 to a Scoping instance.

520

521 CIM\_PolicyRule is the central class of this profile and at least one instance is mandatory.

522 CIM\_PolicyRule instances represent 'If <condition expression> then {<action sequence>}' semantics.

523 The condition expression and action sequence of a CIM\_PolicyRule may be specified implicitly, explicitly,  
524 or both.

525 Each instance of CIM\_PolicySetValidityPeriod creates an implicit condition on the referenced  
526 CIM\_PolicyRule. Evaluation of that rule is conditional on the referenced CIM\_PolicyTimePeriodCondition.

527

528 The set of individual terms of an explicit condition expression are represented by CIM\_PolicyCondition  
529 instances that are aggregated by CIM\_PolicyConditionInPolicyRule instances.

530 The ConditionListType property of a CIM\_PolicyRule instance together with the GroupNumber and  
531 ConditionNegated properties of each CIM\_PolicyConditionInPolicyRule aggregation instance collectively  
532 specify an explicit condition expression.

533 The set of terms is divided into one or more subsets. Each subset is defined by a different value for the  
534 GroupNumber property.

535

536

537 A CIM\_PolicyRule may also be associated with a set of CIM\_PolicyTimePeriodCondition instances that  
538 are each aggregated by an instance of CIM\_PolicySetValidityPeriod. Collectively, this set specifies the  
539 schedule according to which the CIM\_PolicyRule is active and inactive.

540

541 An instance of CIM\_PolicyRule may use a rule set defined by instances of the CIM\_PolicySetComponent  
542 association to associate CIM\_PolicyRule instances. Such a rule is known as a composite rule.

543 The implicit and explicit conditions of the aggregating rule apply to each of the subordinate rules.

544 Any side effects of condition evaluation or the execution of actions shall not affect the result of the  
545 evaluation of other conditions evaluated. That is, the conditions in the composite rule may be evaluated in  
546 any order before applying the priority and determining which actions are to be executed.

547 If the conditions of the composite rule are met, its action sequence shall be executed after all applicable  
548 actions of subordinate rules are executed.

549 The Priority property of CIM\_PolicySetComponent defines the order of execution of CIM\_PolicyRule  
550 instances in a rule set. The value of Priority shall be a unique, non-negative integer value, relative to all  
551 Priority values in the rule set. The actions of an aggregated CIM\_PolicyRule instance with a numerically  
552 higher value of Priority shall be executed before the actions of an aggregated instance with a lower  
553 Priority value.

554 PolicyDecisionStrategy is a conditional property that shall be used in a composite rule to define the  
555 evaluation method used for policies contained in its rule set. The PolicyDecisionStrategy property is  
556 conditional on instance of CIM\_PolicyRule being referenced by an instance of CIM\_PolicySetComponent.

557

558 CIM\_PolicyRuleInSystem instances are a mandatory scoping association that links a CIM\_PolicyRule  
559 instance to the CIM\_System instance in whose scope the rule is defined.

560

561 An instance of CIM\_PolicySetAppliedToElement may be used to define what managed elements can be  
562 affected by the successful execution of the policy. A specialization of this profile may use other  
563 mechanism, like CQL or CPL, to define the managed elements that can be affected by the policy. If a  
564 specialization of this profile relies on this association, then that profile shall define how the association is  
565 used to define the set of managed element affected by policy, accounting for the interaction of that  
566 definition with other set definition mechanisms like CPL if used.

567

568 A CIM\_PolicySetComponent instance is an optional aggregation used to create a composite rule. The set  
569 of aggregated CIM\_PolicyRule instances forms a rule set that is evaluated in the order specified by the  
570 mandatory Priority property.

571 The Priority property of CIM\_PolicySetComponent is a mandatory property defines the order of execution  
572 of CIM\_PolicyRule instances in a rule set.

573

574 An instance of CIM\_PolicySetInRoleCollection is an optional aggregation used to define a set of  
575 CIM\_PolicyRule instances that are necessary to satisfy a policy role defined by the referenced  
576 CIM\_PolicyRoleCollection instance.

577

578 An instance of the optional CIM\_PolicySetValidityPeriod aggregation associates an instance of  
579 CIM\_PolicyTimePeriodCondition with an instance of CIM\_PolicySet.

580 A Time Period may be aggregated by multiple CIM\_PolicySets. A set that does not point to a  
581 CIM\_PolicyTimePeriodCondition through this association, from the scheduling perspective, is always in a  
582 valid time period.

583

584 Each instance of this optional class provides a means of representing the time periods.

585 One use is to govern when an instance of CIM\_PolicyRule is active when aggregated through  
586 CIM\_PolicySetValidityPeriod. At all times that fall outside these time periods, the aggregating  
587 CIM\_PolicyRule has no effect. An instance of CIM\_PolicyRule is active at all times if it does not  
588 aggregate any instances of CIM\_PolicyTimePeriodCondition.

589 A second use of CIM\_PolicyTimePeriodCondition is as an explicit term in a condition expression of an  
590 instance of CIM\_PolicyRule or CIM\_CompoundPolicyCondition. Using CIM\_PolicyTimePeriodCondition in  
591 this way enables the inclusion of time-based criteria in the condition expressions.

592 The TimePeriod property shall match the pattern  
593 `"^[0123456789]{8}T[0123456789]{6}/[0123456789]{8}T[0123456789]{6}$"`.

594 The value of the optional TimePeriod property is formatted as a string that represents a start date and  
595 time, in which the character "T" indicates the beginning of the time portion, followed by the slash character  
596 (/), followed by a similar string that represents an end date and time. The first date indicates the beginning  
597 of the range; the second date indicates the end. The second date and time shall be later than the first.  
598 Date/times are expressed as substrings of the form `yyyymmddThhmmss`. For example,  
599 `20000101T080000/20000131T120000` defines January 1, 2000, 0800 through January 31, 2000, noon.

600 In the following special cases, one of the date/time strings may be replaced with a special string defined  
601 in RFC 2445:

- 602 • If the first date/time is replaced with the string 'THISANDPRIOR', the property indicates that a  
603 PolicySet is valid [from now] until the date/time that appears after the slash character (/).
- 604 • If the second date/time is replaced with the string 'THISANDFUTURE', the property indicates  
605 that a PolicySet becomes valid on the date/time that appears before the slash character (/) and  
606 remains valid from that point on.

607 When the TimePeriod property is NULL, the CIM\_PolicyTimePeriodCondition shall evaluate to TRUE.

608 The value of the optional MonthOfYearMask property refines the valid time period that is defined by the  
609 TimePeriod property by explicitly specifying the months during which the condition may evaluate to  
610 TRUE. This property is formatted as an octet string, structured as follows:

- 611 • A 4-octet length field, indicating the length of the entire octet string. This field is always set to  
612 `0x00000006` for this property.
- 613 • A 2-octet field consisting of 12 bits identifying the 12 months of the year, beginning with January  
614 and ending with December, followed by 4 bits that are always set to 0. For each month, the  
615 value 1 indicates that the policy is valid for that month, and the value 0 indicates that it is not  
616 valid.

617 The MonthOfYearMask value `0x000000060830`, for example, indicates that the condition evaluates to  
618 TRUE only in the months May, November, and December.

619 If MonthOfYearMask is not specified, or the value is set to NULL, all months of the year are specified for  
620 the purpose of evaluating the condition (effectively a value of `0x00000006FFF0`).

621 The value of the optional DayOfMonthMask refines the valid time period that is defined by the TimePeriod  
622 property by explicitly specifying the days of the month during which the condition may evaluate to TRUE.  
623 This property is formatted as an octet string, structured as follows:

- 624 • A 4-octet length field, indicating the length of the entire octet string. This field is always set to  
625 0x0000000C for this property.
- 626 • An 8-octet field consisting of 31 bits identifying the days of the month counting from the  
627 beginning, followed by 31 more bits identifying the days of the month counting from the end,  
628 followed by 2 bits that are always set to 0. For each day, the value 1 indicates that the PolicySet  
629 is valid for that day, and the value 0 indicates that it is not valid.

630 The DayOfMonthMask value 0x0000000C8000000100000000, for example, indicates that a PolicySet is  
631 valid on the first and last days of the month.

632 For months with fewer than 31 days, the digits corresponding to days that the months do not have  
633 (counting in both directions) are ignored.

634 If a DayOfMonthMask value for this property is not provided, or the value is set to NULL, all days of the  
635 month are treated as valid for the purpose of evaluating this condition (effectively a value of  
636 0x0000000CFFFFFFFFFFFFFFFFFC).

637 The value of the optional DayOfWeekMask refines the valid time period that is defined by the TimePeriod  
638 property by explicitly specifying the days of the week during which the condition may evaluate to TRUE.  
639 This property is formatted as an octet string, structured as follows:

- 640 • A 4-octet length field, indicating the length of the entire octet string. This field is always set to  
641 0x00000005 for this property.
- 642 • A 1-octet field consisting of 7 bits identifying the 7 days of the week, beginning with Sunday and  
643 ending with Saturday, followed by 1 bit that is always set to 0. For each day of the week, the  
644 value 1 indicates that the PolicySet is valid for that day, and the value 0 indicates that it is not  
645 valid.

646 The DayOfWeekMask value 0x000000057C, for example, indicates that a PolicySet is valid Monday  
647 through Friday.

648 If a DayOfWeekMask value for this property is not provided, or the value is set to NULL, all days of the  
649 week are treated as valid for the purpose of evaluating this condition (effectively a value of  
650 0x00000005FE).

651 The TimeOfDayMask property shall match the pattern “^T[0123456789]{6}/T[0123456789]{6} \$”.

652 The value of the optional TimeOfDayMask property refines the valid time period that is defined by the  
653 TimePeriod property by explicitly specifying a range of times in a day during which the condition may  
654 evaluate to TRUE. This property is formatted in the style of RFC 2445: a time string beginning with the  
655 character 'T', followed by the slash character (/), followed by a second time string. The first time indicates  
656 the beginning of the range; the second time indicates the end. Times are expressed as substrings of the  
657 form 'Thhmmss'.

658 The second substring always identifies a later time than the first substring. To allow for ranges that span  
659 midnight, however, the value of the second substring may be smaller than the value of the first substring.  
660 Thus, 'T08000/T21000' identifies the range from 0800 to 2100, while 'T21000/T08000' identifies the  
661 range from 2100 to 0800 the following day.

662 When a range spans midnight, it includes parts of two successive days. When one of these days is also  
663 selected by the MonthOfYearMask, DayOfMonthMask, or DayOfWeekMask property, but the other day is  
664 not, then the condition evaluates to TRUE only during the portion of the range that falls on the selected  
665 day. For example, if the range extends from 2100 to 0800, and the DayOfWeekMask property selects  
666 Monday and Tuesday, the PolicySet is active during the following three intervals:

- 667 • From midnight Sunday until 0800 Monday
- 668 • From 2100 Monday until 0800 Tuesday
- 669 • From 2100 Tuesday until 23:59:59 Tuesday

670 If a TimeOfDayMask value is not provided or the value is set to NULL, the condition may evaluate to  
671 TRUE for all hours of the day (effectively a value of 'T000000/T240000').

672 The value of the optional LocalOrUtcTime property indicates whether the times represented in the  
673 TimePeriod property and in the various Mask properties represent local times or UTC times. There is no  
674 provision for mixing local times and UTC times; the value of this property applies to all of the other time-  
675 related properties. Time periods are synchronized worldwide by using the enumeration value 'UTCTime'.  
676 If the goal is to synchronize worldwide on a particular local time (such as 0300 - 0500 in New York), and if  
677 the TimePeriod property spans a Daylight Saving Time transition in New York, it will be necessary to  
678 create multiple instances of CIM\_PolicyTimePeriodCondition, one based on the offset UTC-0500 for the  
679 part of each year when standard time is used in New York, and one based on the offset UTC-0400 for the  
680 part of each year when Daylight Saving Time is used there.

681

682 Instances of the optional CIM\_ReusablePolicy association indicate that the referenced instance of  
683 CIM\_PolicyRule, CIM\_PolicyAction, or CIM\_PolicyCondition is a member of an instance of  
684 CIM\_ReusablePolicyContainer.

685

686 This optional class represents an administratively defined container for reusable instances of  
687 CIM\_PolicyRule, CIM\_PolicyAction, or CIM\_PolicyCondition. This class does not introduce any additional  
688 properties beyond those in its superclass AdminDomain.

689 Even if an instance of CIM\_PolicyRule and CIM\_PolicyCondition is not associated to an instance of  
690 CIM\_ReusablePolicyContainer through CIM\_ReusablePolicy, the instance of CIM\_PolicyRule or  
691 CIM\_PolicyCondition may be aggregated into one or more instances of CIM\_PolicyRule.

692 Each instance of this class shall use the NameFormat value "ReusablePolicyContainer".

693

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

696

697 The mandatory method CIM\_PolicyRoleCollection.ActivatePolicySet() takes as input a reference to a  
698 CIM\_ManagedElement. The CIM\_ManagedElement shall be a member of the CIM\_PolicyRoleCollection,  
699 associated through an instance of ElementInPolicyRoleCollection. The result of this method, if it is  
700 successfully executed, is that the aggregated CIM\_PolicySets are deployed and enforced for the  
701 referenced element. This is reflected by the instantiation of the CIM\_PolicySetAppliesToElement  
702 association between the referenced element and each instance of CIM\_PolicySet.

703

704

705 The mandatory method CIM\_PolicyRoleCollection.DeactivatePolicySet() takes as input a reference to a  
706 CIM\_ManagedElement. The CIM\_ManagedElement shall be a member of the CIM\_PolicyRoleCollection,  
707 associated through an instance of ElementInPolicyRoleCollection. The result of this method, if it is

708 successfully executed, is that the aggregated CIM\_PolicySets are not enforced for the referenced  
709 element. This is reflected by the removal of the CIM\_PolicySetAppliesToElement association instance, if  
710 one exists, between the referenced element and each collected CIM\_PolicySet.

711  
712 Support for operations for each profile class (including associations) is specified in the following  
713 subclauses. Each subclause includes either the statement “All operations are supported as described by  
714 [DSP0200 version 1.2](#)” or a table listing all the operations that are not supported by this profile or where  
715 the profile requires behavior other than that described by [DSP0200 version 1.2](#).

716 The default list of operations is as follows:

- 717 • GetInstance
- 718 • EnumerateInstances
- 719 • EnumerateInstanceNames
- 720 • Associators
- 721 • AssociatorNames
- 722 • References
- 723 • ReferenceNames

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

726  
727 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

728  
729 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

730  
731 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

732  
733 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

734  
735 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

736  
737 Table 2 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
738 shall not be supported.

739

EnumerateInstances	Unspecified	None
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

740  
741 Table 3 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
742 shall not be supported.

743

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



744

745 Table 4 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
746 shall not be supported.

747

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

748

749 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

750

751 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

752

753 Table 5 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
754 shall not be supported.

755

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

756

757

758 Table 6 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
759 shall not be supported.

760

Operation	Requirement	Messages
EnumerateInstances	Unspecified	None
EnumerateInstanceNames	Unspecified	None
Associators	Unspecified	None

AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

761  
762 Table 7 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
763 shall not be supported.

764

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

765  
766 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

767  
768  
769 Table 8 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
770 shall not be supported.

771

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

772

773

774 Table 9 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
775 shall not be supported.

776

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

777

778 Table 10 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
779 or shall not be supported.

780

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

781

782 Table 11 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
783 or shall not be supported.

784

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

785

786 Table 12 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#) or  
787 shall not be supported.

788

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

789

790 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

791

792

793 Table 13 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
794 or shall not be supported.

795

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

796

797 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

798

799

800 This section contains object diagrams and use cases for the *Policy Profile*.

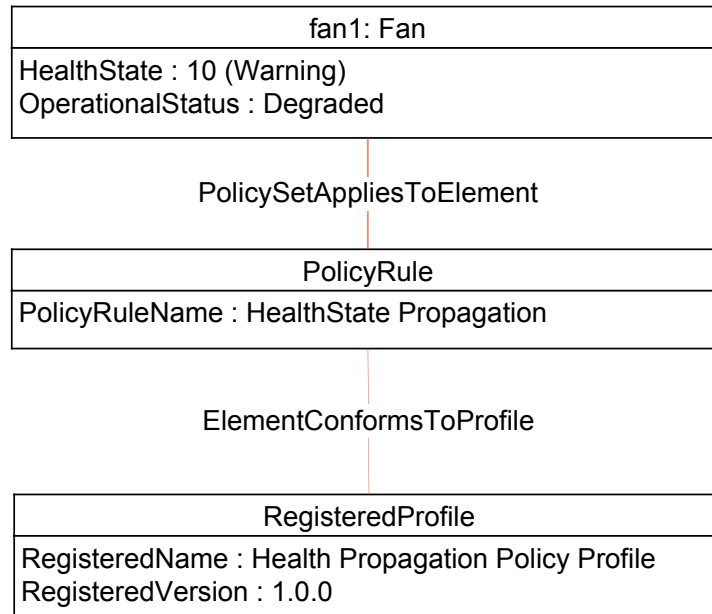
801

802 Figure 2 and Figure 3 represent possible policy definitions in a specialized policy profile. The two  
803 approaches represented are the Blackbox approach in which the specialized profile defines a policy

804 algorithm, policy name, and policy registered profile. Figure 3 represents taking the definition a step  
805 further and defines the actual CQL or CIM-SPL statements that represent the policy algorithm.

806

807 Figure 2 is an example of a Blackbox definition of CIM\_PolicyRule in which the rule represents a policy  
808 algorithm that has been defined in a specialized policy profile but does not have specific CQL or CIM-SPL  
809 defined.

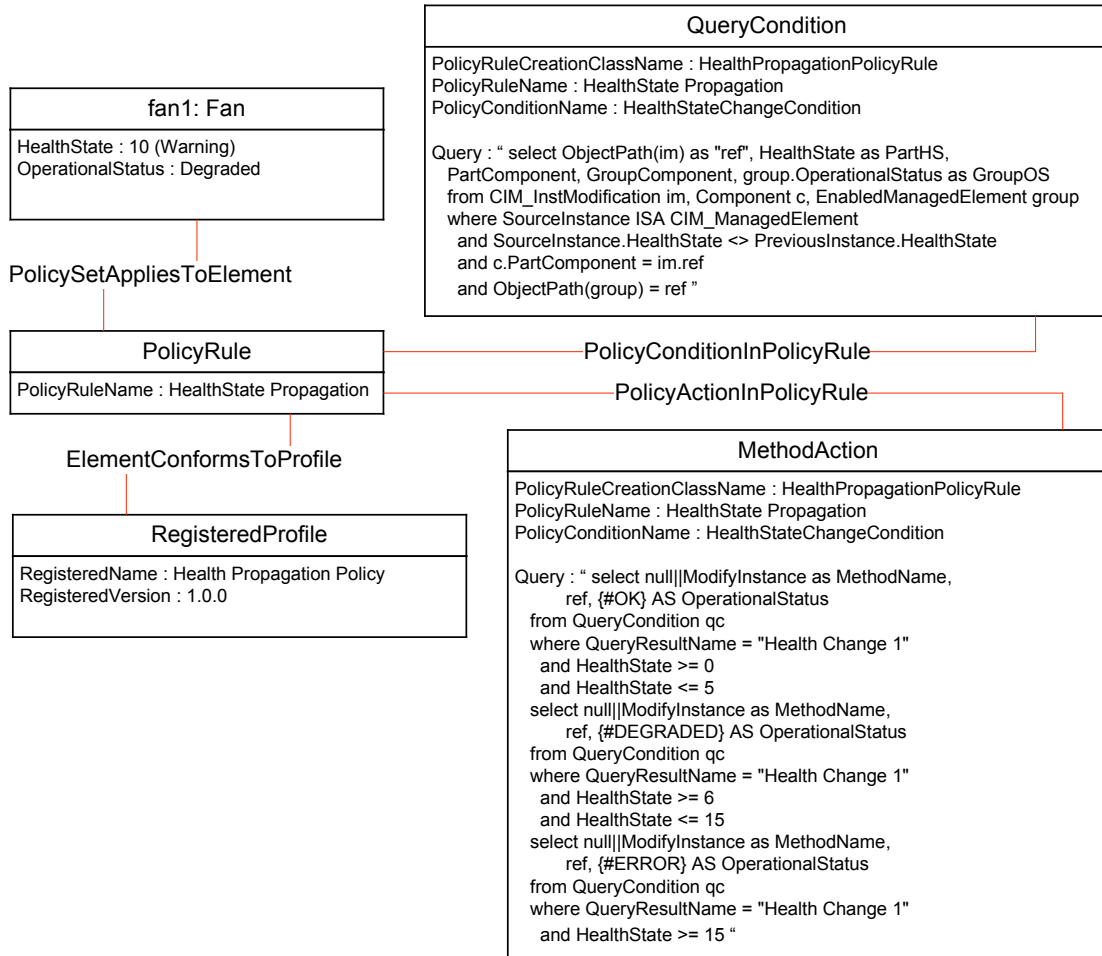


810

811

812

813 Figure 3 is an example of the CIM\_PolicyRule, CIM\_PolicyCondition, CIM\_PolicyAction, and  
 814 CIM\_RegisteredProfile classes instantiated according to a specialized whitebox policy profile that defines  
 815 policy for Health State to OperationalStatus propagation within a class.



816

817

818

819 Table 14 shows the requirements for the CIM Elements in this profile. Instances of the following CIM  
 820 Elements shall be implemented as described in Table 14. Sections 7 (“Implementation”) and 8  
 821 (“Methods”) may impose additional requirements on these elements.

822

CIM_QueryCondition	Optional	See section 7.1 and 10.1
CIM_MethodAction	Optional	See section 7.2 and 10.2

CIM_PolicyRule	Mandatory	See section 7.14 and 10.3.
CIM_PolicyCondition	Optional	See section 7.9 and 10.4.
CIM_PolicyAction	Optional	See section 7.6 and 10.5.
CIM_PolicySetAppliesToElement	Optional	See section 7.16 and 10.6
CIM_PolicyConditionInPolicyRule	Optional	See section 7.11 and 10.7
CIM_PolicyActionInPolicyRule	Optional	See section 7.8 and 10.8
CIM_CompoundPolicyAction	Optional	See section 7.1 and 10.9
CIM_CompoundPolicyCondition	Optional	See section 7.4 and 10.10
CIM_ElementInPolicyRoleCollection	Optional	See section 7.5 and 10.11
CIM_PolicyActionInPolicyAction	Optional	See section 7.7 and 10.12
CIM_PolicyConditionInPolicyCondition	Optional	See section 7.10 and 10.13
CIM_PolicyRoleCollection	Optional	See section 7.12 and 10.14
CIM_PolicyRoleCollectionInSystem	Conditional	See section 7.13 and 10.15
CIM_PolicyRuleInSystem	Mandatory	See section 7.15 and 10.16
CIM_PolicySetComponent	Optional	See section 7.17 and 10.17
CIM_PolicySetInRoleCollection	Optional	See section 7.18 and 10.18
CIM_PolicySetValidityPeriod	Optional	See section 7.19 and 10.19
CIM_PolicyTimePeriodCondition	Optional	See section 7.20 and 10.20
CIM_ReusablePolicy	Optional	See section 7.21 and 10.21
CIM_ReusablePolicyContainer	Optional	See section 7.22 and 10.22
No Indications defined in this version		

823

824 CIM\_QueryCondition defines the criteria for generating a set of query results that are accessible to other  
 825 QueryConditions or MethodActions of the same PolicyRule. Table 15 contains the requirements for  
 826 elements of this class.

827

SystemCreatonClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyConditionName	Mandatory	Key
QueryResultName	Mandatory	This property shall be treated as a class name in a query statement.
Query	Mandatory	This property shall be a query expression that may be evaluated and that defines the query results that may be generated.
QueryLanguage	Mandatory	This property shall be the language defined by DSP0200. The language shall be the one in which the query string is expressed.
Trigger		This property shall be a Boolean condition and there shall be no more than one QueryCondition with this property = true associated with a particular Policy.

828

829 CIM\_MethodAction is a CIM\_PolicyAction that MAY invoke methods as defined by a query. Table 16  
 830 contains the requirements for elements of this class.

831

SystemCreatonClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyActionName	Mandatory	Key
InstMethodCallName	Mandatory	This property shall be treated as a class name in a query statement.
Query	Mandatory	This property shall be a query expression that defines the method to invoke and its input parameters.



QueryLanguage	Mandatory	This property shall be the language defined by DSP0200. The language shall be the one in which the query string is expressed.

832

833

834 CIM\_PolicyRule represents the policy rule applied to the associated CIM\_ManagedElement class. Table  
835 17 contains the requirements for elements of this class.

836

SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
PolicyDecisionStrategy	Conditional	See section 7.14.3
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

837

838 CIM\_PolicyCondition represents a specific condition defined in CQL or CIM-SPL for triggering a related  
839 CIM\_PolicyAction. Table 18 contains the requirements for elements of this class.

840

SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
PolicyConditionName	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

841

842 CIM\_PolicyAction represents a specific action defined in CQL or CIM-SPL to be invoked when a specific  
843 policy condition is triggered. Table 19 contains the requirements for elements of this class.

844

SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
PolicyActionName	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

845

846

847 CIM\_PolicySetAppliesToElement associates the CIM\_PolicySet instance with the CIM\_ManagedElement  
848 instance whose state is evaluated by policy conditions and may be affected by the successful execution  
849 of policy actions. The managed element may be an aggregation point for several other managed element.  
850 In this role, the existence of this association to this aggregation point represents the set of instances  
851 affected and evaluated by the policy.

852 Table 20 contains the requirements for elements of this class.

853

PolicySet	Mandatory	This property shall be a reference to the instance of CIM_PolicySet. Cardinality *
ManagedElement	Mandatory	This property shall be a reference to the instance of CIM_ManagedElement. Cardinality *

854

855

856

857 CIM\_PolicyConditionInPolicyRule is an optional aggregation that may be used to compound  
858 CIM\_PolicyCondition instances into an instance of CIM\_PolicyRule.

859 Table 21 contains the requirements for elements of this class.

860

861

GroupComponent	Mandatory	This property shall be a reference to the instance of CIM_PolicySetRule.
PartComponent	Mandatory	This property shall be a reference to the instance of CIM_PolicyCondition.
GroupNumber	Optional	This property is an optional property used to produce subsets of the set of condition terms.
ConditionNegated	Optional	See section 7.11

862

863 CIM\_PolicyActionInPolicyRule is an optional aggregation that may be used to explicitly specify that an  
864 action, represented by an instance of CIM\_PolicyAction, shall be executed if the conditions specified for  
865 the referenced instance of CIM\_PolicyRule are met.

866 Table 22 contains the requirements for elements of this class.

867

868

GroupComponent	Mandatory	This property shall be a reference to the instance of CIM_PolicyRule.
ManagedElement	Mandatory	This property shall be a reference to the instance of CIM_PolicyAction.
ActionOrder	Optional	This property is an unsigned integer 'n' that indicates the relative position of a PolicyAction in the sequence of actions associated with a PolicyRule or CompoundPolicyAction.

869

870 CIM\_CompoundPolicyAction is an optional class that may be used to represent an expression consisting  
871 of an ordered sequence of action terms. Each action term is represented as a subclass of the  
872 PolicyAction class. Compound actions are constructed by associating dependent action terms together  
873 using the PolicyActionInPolicyAction aggregation.

874 Table 23 contains the requirements for elements of this class.

875

876

SequencedActions	Optional	See Section 7.3.1
ExecutionStrategy	Optional	See Section 7.3
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
CreationClassName	Mandatory	Key
PolicyActionName	Mandatory	Key

877

878

879 CIM\_CompoundPolicyCondition is an optional class that may be used to represent compound conditions  
880 formed by aggregating simpler policy conditions. Compound conditions are constructed by associating  
881 subordinate condition terms together using the PolicyConditionInPolicyCondition aggregation.

882 Table 24 contains the requirements for elements of this class.

883

884

SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
CreationClassName	Mandatory	Key
PolicyConditionName	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

885

886 CIM\_ElementInPolicyRoleCollection is an optional association that may be used to aggregate zero or  
887 more Managed Element subclass instances into a PolicyRoleCollection object, representing a role played  
888 by these Managed Elements. This Collection indicates that the aggregated PolicySets (aggregated by  
889 CIM\_PolicySetInRoleCollection) MAY BE applied to the referenced elements.

890 Table 25 contains the requirements for elements of this class.

891

892

Collection	Mandatory	Key
Member	Mandatory	Key

893

894

895 CIM\_PolicyActionInPolicyAction is an optional association that may be used to represent the  
896 compounding of policy actions into a higher-level policy action.

897 Table 26 contains the requirements for elements of this class.

898

899

GroupComponent	Mandatory	Key
PartComponent	Mandatory	Key
ActionOrder	Optional	This property is an unsigned integer ' n ' that indicates the relative position of a PolicyAction in the sequence of actions associated with a PolicyRule or CompoundPolicyAction.

900

901

902 CIM\_PolicyConditionInPolicyCondition is an optional association that may be used to aggregate zero or  
903 more instances of the PolicyCondition class to a CompoundPolicyCondition.

904 Table 27 contains the requirements for elements of this class.

905

906

GroupComponent	Mandatory	Key
PartComponent	Mandatory	Key
GroupNumber	Optional	See Section 7.10
ConditionNegated	Optional	See Section 7.10

907

908 CIM\_PolicyRoleCollection is an optional class that may be used to to represent a collection of Managed  
909 Elements that share a common policy role, and the PolicySets that CAN BE applied to those elements.  
910 The PolicyRoleCollection always exists in the context of a System, specified using the  
911 PolicyRoleCollectionInSystem aggregation. The value of the PolicyRole property in this class specifies

912 the role. It is defined as a free-form string. Managed Elements that share the role defined in this collection  
 913 are aggregated into the Collection via the ElementInPolicyRoleCollection association.

914 Table 28 contains the requirements for elements of this class.

915

916

PolicyRole	Mandatory	See Section 7.12
InstanceID	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

917

918 CIM\_PolicyRoleCollectionInSystem is a conditional association that shall be used to establish a  
 919 relationship between an instance of CIM\_PolicyRoleCollection and an "owning" CIM\_System instance,  
 920 such as an AdminDomain or ComputerSystem.

921 Table 29 contains the requirements for elements of this class.

922

923

Antecedent	Mandatory	Key
Dependent	Mandatory	Key

924

925 CIM\_PolicyRuleInSystem is a mandatory association that shall be used to link a PolicyRule to the System  
 926 in whose scope the Rule is defined.

927 Table 30 contains the requirements for elements of this class.

928

929

Antecedent	Mandatory	Key
Dependent	Mandatory	Key

930

931 CIM\_PolicySetComponent is an optional association that may be used to aggregate the instances of the  
 932 subclasses of PolicySet (i.e., PolicyGroups and PolicyRules). Instances are collected in sets that use the  
 933 same decision strategy. They are prioritized relative to each other, within the set, using the Priority  
 934 property of this aggregation. Together, the PolicySet.PolicyDecisionStrategy and PolicySet  
 935 Component.Priority properties determine the processing for the groups and rules contained in a  
 936 PolicySet. A larger priority value represents a higher priority. Note that the Priority property MUST have a

937 unique value when compared with others defined for the same aggregating PolicySet. Thus, the  
938 evaluation of rules within a set is deterministically specified.

939 Table 31 contains the requirements for elements of this class.

940

941

GroupComponent	Mandatory	Key
PartComponent	Mandatory	Key
Priority	Mandatory	See section 7.17

942

943 CIM\_PolicySetInRoleCollection is an optional association that may be used to aggregate zero or more  
944 PolicyRules and PolicyGroups (i.e., the subclasses of PolicySet) into a PolicyRoleCollection object,  
945 representing a role supported/enforced by the PolicySet.

946 Table 32 contains the requirements for elements of this class.

947

948

Collection	Mandatory	Key
Member	Mandatory	Key

949

950 CIM\_PolicySetValidityPeriod is an optional association that may be used to aggregate scheduled  
951 activation and deactivation of a PolicySet. A PolicySet is considered "active" if it is both "Enabled" and in  
952 a valid time period. If a PolicySet is associated with multiple policy time periods via this association, then  
953 the Set is in a valid time period if at least one of the time periods evaluates to TRUE. If a PolicySet is  
954 contained in another PolicySet via the PolicySetComponent aggregation (e.g., a PolicyRule in a  
955 PolicyGroup), then the contained PolicySet (e.g., PolicyRule) is in a valid period if at least one of the  
956 aggregate's PolicyTimePeriodCondition instances evaluates to TRUE and at least one of its own  
957 PolicyTimePeriodCondition instances also evaluates to TRUE. (In other words, the  
958 PolicyTimePeriodConditions are ORed to determine whether the PolicySet is in a valid time period and  
959 then ANDed with the ORed PolicyTimePeriodConditions of each of PolicySet instances in the  
960 PolicySetComponent hierarchy to determine if the PolicySet is in a valid time period and, if also  
961 "Enabled", therefore, active, i.e., the hierarchy ANDs the ORed PolicyTimePeriodConditions of the  
962 elements of the hierarchy. A Time Period may be aggregated by multiple PolicySets. A Set that does not  
963 point to a PolicyTimePeriodCondition via this association, from the point of view of scheduling, is always  
964 in a valid time period.

965 Table 33 contains the requirements for elements of this class.

966

967

GroupComponent	Mandatory	Key
PartComponent	Mandatory	Key

968

969 CIM\_PolicyTimePeriodCondition is an optional class that may be used to represent the time periods  
 970 during which a PolicySet is valid, i.e., active. At all times that fall outside these time periods, the PolicySet  
 971 has no effect.

972 Table 34 contains the requirements for elements of this class.

973

974

TimePeriod	Optional	See Section 7.20
MonthOfYearMask	Optional	See Section 7.20
DayOfMonthMask	Optional	See Section 7.20
DayOfWeekMask	Optional	See Section 7.20
TimeOfDayMask	Optional	See Section 7.20
LocalOrUtcTime	Optional	See Section 7.20
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyRuleCreationClassName	Mandatory	Key
PolicyRuleName	Mandatory	Key
CreationClassName	Mandatory	Key
PolicyConditionName	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

975

976 CIM\_ReusablePolicy is an optional association that may be used to establish dependency relationships  
 977 between Policies and the Systems that host them.

978 Table 35 contains the requirements for elements of this class.

979



980

Antecedent	Mandatory	Key
Dependent	Mandatory	Key

981

982

983 CIM\_ReusablePolicyContainer is an optional class that may be used to represente an administratively  
984 defined container for reusable policy-related information. This class does not introduce any additional  
985 properties beyond those in its superclass AdminDomain. It does, however, participate in a unique  
986 association for containing policy elements. An instance of this class uses the NameFormat value  
987 "ReusablePolicyContainer".

988 Table 36 contains the requirements for elements of this class.

989

990

CreationClassName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

991

992

993  
994  
995  
996


997

998  
999  
1000  
1001

1002 The authors wish to acknowledge the following people.

1003

1004 • Jianwen Yin – Dell

1005

1006 • George Ericson – EMC

1007 • Jon Hass – Dell

1008 • Jianwen Yin – Dell

1009 • Steve Hand – Symantec Corp.

1010 • Aaron Merkin – IBM

1011