



1

2

3

4

**Document Number: DSP1039**

**Date: 2008-10-31**

**Version: 1.0.0**

## 5 **Role Based Authorization Profile**

6 **Document Type: Specification**

7 **Document Status: Final**

8 **Document Language: E**

9

10 Copyright notice

11 Copyright © 2008 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 for uses consistent with this purpose, provided that correct attribution is given. As DMTF  
15 specifications may be revised from time to time, the particular version and release date should always be  
16 noted.

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

30

# Contents

32	Foreword .....	6
33	Introduction .....	7
34	1 Scope .....	9
35	2 Normative References.....	9
36	2.1 Approved References .....	9
37	2.2 References under Development .....	9
38	2.3 Other References.....	9
39	3 Terms and Definitions .....	9
40	4 Symbols and Abbreviated Terms .....	11
41	5 Synopsis .....	12
42	6 Description .....	12
43	6.1 Role Authorization Service: CIM_RoleBasedAuthorizationService .....	13
44	6.2 Authorized Roles and Privileges: CIM_Role and CIM_Privilege .....	13
45	6.3 Security Principal: CIM_Identity .....	14
46	6.4 Privilege Management .....	14
47	7 Implementation.....	15
48	7.1 Modeling the Authorized Role.....	15
49	7.2 Authorized Role Management .....	18
50	7.3 Authorized Role Membership of Security Principal .....	19
51	7.4 Privilege Management Capability .....	20
52	8 Methods.....	21
53	8.1 CIM_RoleBasedAuthorizationService.CreateRole( ) .....	21
54	8.2 CIM_RoleBasedAuthorizationService.DeleteRole( ) .....	22
55	8.3 CIM_RoleBasedAuthorizationService.ModifyRole( ) .....	24
56	8.4 CIM_RoleBasedAuthorizationService.AssignRoles( ) .....	25
57	8.5 CIM_RoleBasedAuthorizationService.ShowAccess( ) .....	26
58	8.6 CIM_RoleBasedAuthorizationService.ShowRoles( ) .....	27
59	8.7 Profile Conventions for Operations.....	28
60	8.8 CIM_ConcreteDependency .....	29
61	8.9 CIM_ElementCapabilities .....	29
62	8.10 CIM_HostedService .....	30
63	8.11 CIM_MemberOfCollection .....	30
64	8.12 CIM_OwningCollectionElement .....	30
65	8.13 CIM_Privilege.....	30
66	8.14 CIM_RoleBasedManagementCapabilities .....	31
67	8.15 CIM_Role .....	31
68	8.16 CIM_RoleBasedAuthorizationService.....	31
69	8.17 CIM_RoleLimitedToTarget.....	31
70	8.18 CIM_ServiceAffectsElement .....	31
71	8.19 CIM_ServiceServiceDependency .....	32
72	9 Use Cases .....	32
73	9.1 Profile Registration.....	32
74	9.2 Minimal Instantiation of the Profile.....	33
75	9.3 Evaluating Scope and Privileges .....	33
76	9.4 Scope of the Role and Privileges for a Managed Element.....	36
77	9.5 Service Processor Roles Use Cases.....	39
78	9.6 Determine the Roles Managed by a Service .....	42
79	9.7 Determine Candidate Roles for a Security Principal .....	42
80	9.8 Determine the Roles to Which a Security Principal Is Currently Assigned.....	42
81	9.9 Determine the Roles that Scope a Managed Element .....	43
82	9.10 Determine the Current Privileges of a Security Principal for a Managed Element.....	43

83 9.11 Modify a Single Privilege of an Existing Role ..... 44

84 9.12 Create a New Role..... 44

85 9.13 Determine Whether Privilege Management Is Supported for a Principal ..... 44

86 9.14 Determine Whether One-to-One Privilege Management Is Supported for an Account..... 45

87 9.15 Assign Custom Privileges to an Identity ..... 45

88 10 CIM Elements ..... 46

89 10.1 CIM\_ConcreteDependency (Privilege) ..... 46

90 10.2 CIM\_ConcreteDependency (Role)..... 47

91 10.3 CIM\_ElementCapabilities ..... 47

92 10.4 CIM\_HostedService ..... 47

93 10.5 CIM\_MemberOfCollection (Privilege) ..... 48

94 10.6 CIM\_MemberOfCollection (Identity) ..... 48

95 10.7 CIM\_OwningCollectionElement ..... 48

96 10.8 CIM\_Privilege..... 49

97 10.9 CIM\_RoleBasedManagementCapabilities ..... 49

98 10.10 CIM\_RegisteredProfile ..... 49

99 10.11 CIM\_Role ..... 50

100 10.12 CIM\_RoleBasedAuthorizationService..... 50

101 10.13 CIM\_RoleLimitedToTarget..... 50

102 10.14 CIM\_ServiceAffectsElement – CIM\_Role ..... 51

103 10.15 CIM\_ServiceAffectsElement – CIM\_Privilege ..... 51

104 10.16 CIM\_ServiceServiceDependency ..... 51

105 ANNEX A (informative) Change Log..... 53

106 ANNEX B (informative) Acknowledgements ..... 54

107

108 **Figures**

109 Figure 1 – Role Based Authorization Profile: Class Diagram ..... 13

110 Figure 2 – Profile Registration..... 32

111 Figure 3 – Minimal Instantiation ..... 33

112 Figure 4 – Cumulative Role Privilege Example..... 34

113 Figure 5 – Scope of the Roles ..... 37

114 Figure 6 – Fixed Accounts with Role Membership Privilege Management ..... 38

115 Figure 7 – Fixed Accounts with Individual Account Privilege Management ..... 39

116 Figure 8 – IPMI Service Processor with Role Management ..... 40

117 Figure 9 – IPMI Service Processor with Role Management ..... 41

118

119 **Tables**

120 Table 1 – Referenced Profiles ..... 12

121 Table 2 – Containment Relationships ..... 16

122 Table 3 – CIM\_RoleBasedAuthorizationService.CreateRole( ) Method: Return Code Values ..... 22

123 Table 4 – CIM\_RoleBasedAuthorizationService.CreateRole( ) Method: Parameters ..... 22

124 Table 5 – CIM\_RoleBasedAuthorizationService.DeleteRole( ) Method: Return Code Values..... 23

125 Table 6 – CIM\_RoleBasedAuthorizationService.DeleteRole( ) Method: Parameters ..... 23

126 Table 7 – CIM\_RoleBasedAuthorizationService.ModifyRole( ) Method: Return Code Values ..... 24

127 Table 8 – CIM\_RoleBasedAuthorizationService.ModifyRole( ) Method: Parameters ..... 24

128 Table 9 – CIM\_RoleBasedAuthorizationService.AssignRoles( ) Method: Return Code Values ..... 25

129 Table 10 – CIM\_RoleBasedAuthorizationService.AssignRoles( ) Method: Parameters ..... 25

130 Table 11 – CIM\_RoleBasedAuthorizationService.ShowAccess( ) Method: Return Code Values..... 26

131 Table 12 – CIM\_RoleBasedAuthorizationService.ShowAccess( ) Method: Parameters..... 26

132 Table 13 – CIM\_RoleBasedAuthorizationService.ShowRoles( ) Method: Return Code Values ..... 28

133 Table 14 – CIM\_RoleBasedAuthorizationService.ShowRoles( ) Method: Parameters ..... 28

134 Table 15 – Operations: CIM\_ConcreteDependency ..... 29

135 Table 16 – Operations: CIM\_ElementCapabilities ..... 29

136 Table 17 – Operations: CIM\_HostedService ..... 30

137 Table 18 – Operations: CIM\_MemberOfCollection ..... 30

138 Table 19 – Operations: CIM\_OwningCollectionElement ..... 30

139 Table 20 – Operations: CIM\_Privilege ..... 30

140 Table 21 – Operations: CIM\_RoleLimitedToTarget ..... 31

141 Table 22 – Operations: CIM\_ServiceAffectsElement ..... 31

142 Table 23 – Operations: CIM\_ServiceServiceDependency ..... 32

143 Table 24 – CIM Elements: Role Based Authorization Profile ..... 46

144 Table 25 – Class: CIM\_ConcreteDependency (Privilege) ..... 46

145 Table 26 – Class: CIM\_ConcreteDependency (Role)..... 47

146 Table 27 – Class: CIM\_ElementCapabilities..... 47

147 Table 28 – Class: CIM\_HostedService ..... 47

148 Table 29 – Class: CIM\_MemberOfCollection (Privilege) ..... 48

149 Table 30 – Class: CIM\_MemberOfCollection (Identity) ..... 48

150 Table 31 – Class: CIM\_OwningCollectionElement ..... 48

151 Table 32 – Class: CIM\_Privilege ..... 49

152 Table 33 – Class: CIM\_RoleBasedManagementCapabilities ..... 49

153 Table 34 – Class: CIM\_RegisteredProfile ..... 49

154 Table 35 – Class: CIM\_Role ..... 50

155 Table 36 – Class: CIM\_RoleBasedAuthorizationService ..... 50

156 Table 37 – Class: CIM\_RoleLimitedToTarget..... 50

157 Table 38 – Class: CIM\_ServiceAffectsElement ..... 51

158 Table 39 – Class: CIM\_ServiceAffectsElement ..... 51

159 Table 40 – Class: CIM\_ServiceServiceDependency ..... 52

160

161

## Foreword

162 The *Role Based Authorization Profile* (DSP1039) was prepared by the Security Working Group, Server  
163 Management Working Group, and WBEM Infrastructure and Protocols Working Group of DMTF.

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

166

## Introduction

167 This document defines the classes used to describe role-based authorization in a managed system. Also  
168 included are descriptions of the relationship between the authorization and authentication for a managed  
169 system, and the DMTF profile version information. The information in this specification is intended to be  
170 sufficient for a provider or consumer of this data to identify unambiguously the classes, properties,  
171 methods, and values that are mandatory to be instantiated and manipulated to represent and manage  
172 users and groups that are modeled using the DMTF Common Information Model (CIM) core and  
173 extended model definitions.

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





176

# Role Based Authorization Profile

## 177 1 Scope

178 The *Role Based Authorization Profile* extends the management capability of the referencing profiles by  
179 adding the capability to model role-based authorization for a managed system. This profile is intended to  
180 be used for the representation of the authorization on a managed system. This profile is not intended to  
181 serve as a mechanism for the authorization. The relationship between authorization and security  
182 principals of the accounts and groups, as well as the profile's registration for the schema implementation  
183 version information, is also described.

## 184 2 Normative References

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 2.1 Approved References

- 189 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*  
190 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*  
191 DMTF [DSP1000](#), *Management Profile Specification Template*  
192 DMTF [DSP1001](#), *Management Profile Specification Usage Guide*  
193 DMTF [DSP1034](#), *Simple Identity Management Profile 1.0*  
194 DMTF [DSP1033](#), *Profile Registration Profile 1.0*

### 195 2.2 References under Development

- 196 DMTF [DSP0215](#), *Server Management Managed Element Addressing Specification, 1.0.0*

### 197 2.3 Other References

- 198 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,  
199 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>  
200 Unified Modeling Language (UML) from the Open Management Group (OMG), <http://www.uml.org>

## 201 3 Terms and Definitions

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

### 204 3.1

#### 205 can

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

- 207 **3.2**  
208 **cannot**  
209 used for statements of possibility and capability, whether material, physical, or causal
- 210 **3.3**  
211 **conditional**  
212 indicates requirements to be followed strictly to conform to the document when the specified conditions  
213 are met
- 214 **3.4**  
215 **mandatory**  
216 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
217 permitted
- 218 **3.5**  
219 **may**  
220 indicates a course of action permissible within the limits of the document
- 221 **3.6**  
222 **need not**  
223 indicates a course of action permissible within the limits of the document
- 224 **3.7**  
225 **optional**  
226 indicates a course of action permissible within the limits of the document
- 227 **3.8**  
228 **referencing profile**  
229 indicates a profile that owns the definition of this class and can include a reference to this profile in its  
230 "Referenced Profiles" table
- 231 **3.9**  
232 **shall**  
233 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
234 permitted
- 235 **3.10**  
236 **shall not**  
237 indicates requirements to be followed strictly to conform to the document and from which no deviation is  
238 permitted
- 239 **3.11**  
240 **should**  
241 indicates that among several possibilities, one is recommended as particularly suitable, without  
242 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 243 **3.12**  
244 **should not**  
245 indicates that a certain possibility or course of action is deprecated but not prohibited
- 246 **3.13**  
247 **unspecified**  
248 indicates that this profile does not define any constraints for the referenced CIM element or operation

249 **3.14**

250 **Associated Privilege Management Capability**

251 an instance of CIM\_RoleBasedManagementCapabilities describing the capabilities of the mentioned  
252 instance of CIM\_Privilege as described in section 7.4

253 **3.15**

254 **Associated Role Management Capability**

255 an instance of CIM\_RoleBasedManagementCapabilities, which is associated with the instance of  
256 CIM\_RoleBasedAuthorizationService through the CIM\_ElementCapabilities association, which in turn is  
257 associated with the mentioned instance of CIM\_Role through the CIM\_ServiceAffectsElement association

258 **3.16**

259 **Cumulative Privilege**

260 a conceptual instance of CIM\_Privilege that represents rights granted

261 **3.17**

262 **Cumulative Role Privilege**

263 an instance of CIM\_Privilege that is the conceptual representation of all the Granted Privileges and  
264 Denied Privileges that are associated with a particular instance of CIM\_Role

265 **3.18**

266 **Denied Privilege**

267 an instance of CIM\_Privilege with the PrivilegeGranted property set to FALSE that represents the denied  
268 privilege of associated roles

269 **3.19**

270 **Granted Privilege**

271 an instance of CIM\_Privilege with the PrivilegeGranted property set to TRUE that represents the granted  
272 privilege of associated roles

273 **3.20**

274 **Modified Role**

275 an instance of CIM\_Role that is referenced by the Role parameter of the ModifyRole() method

276 **3.21**

277 **Root Instance**

278 an instance of CIM\_ManagedElement that is associated with the instance of CIM\_Role through the  
279 CIM\_RoleLimitedToTarget association and conceptually symbolizes the root of the scope hierarchy for  
280 the CIM\_Role instance

281 **3.22**

282 **Template Privilege**

283 an instance of CIM\_Privilege only to be used by a client as a template for creating new authorized roles  
284 or modifying the existing roles

285 **4 Symbols and Abbreviated Terms**

286 **Experimental Maturity Level**

287  
288 Some of the content considered for inclusion in the *Role Based Authorization Profile* has yet to receive  
289 sufficient review to satisfy the adoption requirements set forth by the Technical Committee within the  
290 DMTF. This content is presented here as an aid to implementers who are interested in likely future  
291 developments within this specification. The content marked experimental may change as implementation

292 experience is gained. There is a high likelihood that it will be included in an upcoming revision of the  
 293 specification. Until that time, it is purely informational, and is clearly marked within the text.  
 294 A sample of the typographical convention for experimental content is included here:

---

295 **EXPERIMENTAL**

296 Experimental content appears here.

297 **EXPERIMENTAL**

---

298

299 **5 Synopsis**

300 **Profile Name:** *Role Based Authorization*

301 **Version:** 1.0.0

302 **Organization:** DMTF

303 **CIM schema version:** 2.20

304 **Central Class:** CIM\_RoleBasedAuthorizationService

305 **Scoping Class:** CIM\_ComputerSystem

306 The *Role Based Authorization Profile* extends the management capability of the referencing profiles by  
 307 adding the capability to authorize the authenticated entities in a managed system.

308 The Central Class of the *Role Based Authorization Profile* shall be CIM\_RoleBasedAuthorizationService.  
 309 The Central Instance shall be an instance of CIM\_RoleBasedAuthorizationService. The Scoping Class  
 310 shall be CIM\_ComputerSystem. The Scoping Instance shall be the instance of CIM\_ComputerSystem  
 311 that is associated with the Central Instance through the CIM\_HostedService association.

312 Table 1 lists the profiles related to the *Role Based Authorization Profile*.

313

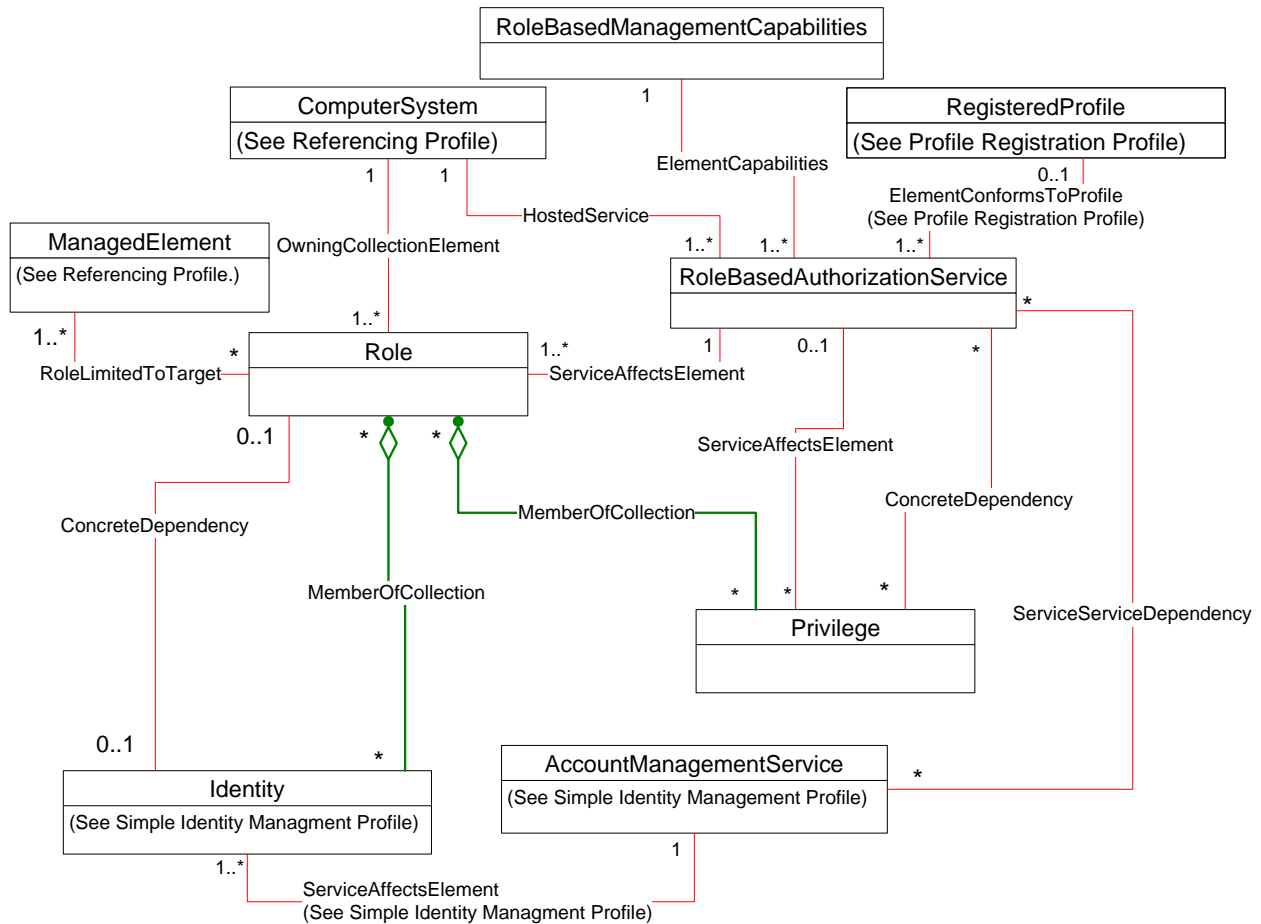
**Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Relationship	Behavior
<i>Simple Identity Management</i>	DMTF	1.0.0	Optional	See section 7.3.
<i>Profile Registration</i>	DMTF	1.0.0	Mandatory	

314 **6 Description**

315 The *Role Based Authorization Profile* describes the properties and methods for role management and  
 316 authorization in a managed system. This profile does not provide a mechanism for an application to verify  
 317 authorization. The CIM instrumentation of this profile is intended to reflect the roles and privileges that are  
 318 available in and enforced by the underlying managed system.

319 Figure 1 represents the class schema for the profile. For simplicity, the prefix *CIM\_* has been removed  
 320 from the names of the classes.



321

322

**Figure 1 – Role Based Authorization Profile: Class Diagram**

323 **6.1 Role Authorization Service: CIM\_RoleBasedAuthorizationService**

324 The ability to manage and configure roles for a managed system is represented by the  
 325 CIM\_RoleBasedAuthorizationService instance. The CIM\_RoleBasedAuthorizationService class is the  
 326 Central Class of the profile and, through extrinsic methods, serves as the interface for a client to request  
 327 deletion and modification of existing roles, creation of new roles, and assignment of roles to security  
 328 principals.

329 **6.2 Authorized Roles and Privileges: CIM\_Role and CIM\_Privilege**

330 The authorized roles on a managed system are represented through instances of CIM\_Role. Rights  
 331 granted to a security principal through membership in a role are represented by instances of  
 332 CIM\_Privilege that are associated with the instance of CIM\_Role through the CIM\_MemberOfCollection  
 333 association.

### 334 6.2.1 Role Privileges

335 When the security principal is a member of an authorized role, the principal is granted the cumulative  
336 privileges of the role. Every authorized role on the managed system can have a set of explicitly granted or  
337 denied privileges. The PrivilegeGranted property of the CIM\_Privilege instance represents whether the  
338 instance of CIM\_Privilege comprises activities that are granted or denied for the role. The Activities,  
339 ActivityQualifiers, and QualifierFormats properties of the CIM\_Privilege instance describe the activities  
340 represented by the privilege.

### 341 6.2.2 Role Scope

342 The scope of the authorized role is the set of managed elements represented by the instances of the  
343 CIM\_ManagedElement subclass, which could be subjected to the activities that make up the privileges of  
344 the authorized role. The scope of the roles authorization is represented by associating the CIM\_Role  
345 instance to instances of CIM\_ManagedElement through the CIM\_RoleLimitedToTarget association. When  
346 the associated CIM\_ManagedElement instance contains or aggregates additional CIM\_ManagedElement  
347 instances, the privileges granted by the role can propagate to the contained or aggregated instances of  
348 CIM\_ManagedElement. This profile does not provide a mechanism for managing whether the privileges  
349 granted by an instance of CIM\_Role for managing an instance of CIM\_ManagedElement are propagated  
350 to aggregated or contained instances of CIM\_ManagedElement. Therefore, privileges granted for  
351 managing or accessing an instance of CIM\_ManagedElement always propagate to the aggregated and  
352 contained instances of CIM\_ManagedElement.

353 The detailed requirements for representing the scope of the authorized role are described in section  
354 7.1.1.

### 355 6.2.3 Cumulative Privileges

356 A security principal is granted rights through role membership to manage or access managed elements  
357 that are within the scope of the role. The Cumulative Privileges granted to a security principal for a  
358 managed element are determined by evaluating the Cumulative Role Privileges for each role of which the  
359 security principal is a member and in whose scope the target managed element lies.

## 360 6.3 Security Principal: CIM\_Identity

361 The CIM\_Identity class represents the security principal for the accounts (CIM\_Account), users  
362 (CIM\_UserContact), and groups (CIM\_Group) as described in the *Simple Identity Management Profile*.  
363 The security principal exists on the managed system and is used to provide the security context under  
364 which the authenticated user and group can act within the managed system. As such, the instantiation of  
365 a CIM\_Identity instance that represents the security principal does not depend on the underlying  
366 authentication of the associated users and groups.

367 CIM\_Identity instances that represent security principals for the accounts, users, and groups can have a  
368 CIM\_MemberOfCollection association to the appropriate CIM\_Role instances. The representation of roles  
369 is described in detail in section 6.2.

## 370 6.4 Privilege Management

371 Two general patterns exist for managing privileges for a security principal. Privileges can be managed  
372 through one or more common roles with well-known, fixed privileges. For example, a system could have  
373 administrator, operator, and read-only roles. The second pattern is the specification of a custom  
374 combination of privileges. These custom privileges can be assigned in two ways. A common role can be  
375 created that has the custom privileges, and then the security principal can be assigned to the role.  
376 Alternatively, each security principal can have a dedicated role, and the custom privileges can be  
377 managed for that role.

378 This profile describes how to use the *Role Based Authorization Profile* to support these two privilege-  
379 management patterns. Two methods can be used. One method uses common roles and manages  
380 privileges for a security principal through membership in one or more roles. The second method uses a  
381 dedicated role for each security principal to enable the management of privileges directly for the principal.  
382 The first method corresponds to the management of privileges (well-known or custom) through  
383 membership in common roles. The second method corresponds to the management of custom privileges  
384 assigned individually to each security principal. Within an implementation, the two methods can be used  
385 simultaneously to model custom and defined roles.

386 When referencing an instance of `CIM_Role`, `CIM_ConcreteDependency` is used to indicate that the  
387 `CIM_Role` instance is dedicated to managing the privileges of the referenced `CIM_Identity`.

388 The `CIM_ServiceServiceDependency` association is used to associate instances of  
389 `CIM_AccountManagementService` with instances of `CIM_RoleBasedAuthorizationService`. This  
390 association indicates that security principals managed by the instance of  
391 `CIM_AccountManagementService` can be assigned to roles managed by the instance of  
392 `CIM_RoleBasedAuthorizationService`.

## 393 **7 Implementation**

394 This section details the requirements related to the arrangement of instances and their properties for  
395 implementations of this profile.

### 396 **7.1 Modeling the Authorized Role**

397 The implementation shall instantiate at least one instance of `CIM_Role` that represents an authorized role  
398 and at least one instance of `CIM_RoleBasedAuthorizationService`.

399 Instances of `CIM_Role` shall be associated to an instance of `CIM_RoleBasedAuthorizationService`  
400 through `CIM_ServiceAffectsElement` associations.

401 Each instance of `CIM_RoleBasedAuthorizationService` shall be associated to only one instance of  
402 `CIM_ComputerSystem` through the `CIM_HostedService` association. This instance of  
403 `CIM_ComputerSystem` shall be the Scoping Instance.

404 Each `CIM_Role` instance shall be associated to only one instance of `CIM_ComputerSystem`, through the  
405 `CIM_OwningCollectionElement` association.

406 Exactly one instance of `CIM_RoleBasedManagementCapabilities` shall be associated with the  
407 `CIM_RoleBasedAuthorizationService` instance through the `CIM_ElementCapabilities` association.

#### 408 **7.1.1 Scope of the Authorized Role**

409 Privileges granted by an instance of `CIM_Role` shall propagate from containing or aggregating instances  
410 of `CIM_ManagedElement` to the contained or aggregated instances of `CIM_ManagedElement`.

411 Each instance of `CIM_Role` shall be referenced by at least one instance of `CIM_RoleLimitedToTarget`.  
412 The `CIM_RoleLimitedToTarget` association explicitly places the referenced instance of  
413 `CIM_ManagedElement` into the scope of the `CIM_Role` instance. Additional instances of  
414 `CIM_ManagedElement` may be implicitly within the scope of the `CIM_Role` instance.

415 Table 2 identifies common containment and aggregation associations that are used to determine if an  
416 instance of `CIM_ManagedElement` is implicitly within the scope of an instance of `CIM_Role`.

417

**Table 2 – Containment Relationships**

<b>Container Class (REF role)</b>	<b>Association Class</b>	<b>Contained Class (REF role)</b>
CIM_ManagedElement (GroupComponent)	CIM_Component	CIM_ManagedElement (PartComponent)
CIM_ManagedElement (Antecedent)	CIM_Dependency	CIM_ManagedElement (Dependent)
CIM_Collection (Collection)	CIM_MemberOfCollection	CIM_ManagedElement (Member)
CIM_ManagedElement (OwningElement)	CIM_OwningCollectionElement	CIM_Collection (OwnedElement)
CIM_RecordLog (Log)	CIM_LogManagesRecord	CIM_LogRecord (Record)
CIM_System (System)	CIM_InstalledSoftwareIdentity	CIM_SoftwareIdentity (InstalledSoftware)

418 **7.1.1.1 Managed Element within Role’s Scope**

419 This section defines the algorithm used to determine whether an instance of CIM\_ManagedElement is  
 420 within the scope of an instance of CIM\_Role.

421 An instance of CIM\_ManagedElement shall be in the scope of an instance of CIM\_Role if

- 422 1) The instance of CIM\_ManagedElement is associated with the instance of CIM\_Role through the  
 423 CIM\_RoleLimitedToTarget association.
- 424 2) The instance of CIM\_ManagedElement is referenced by an instance of an association class  
 425 specified in the "Association Class" column of Table 2 where a reference to the instance of  
 426 CIM\_ManagedElement is the value of the property specified in the "Contained Class" column of  
 427 Table 2 and the instance of CIM\_ManagedElement referenced by the property specified in the  
 428 "Container Class" column of Table 2 is in the scope of the instance of CIM\_Role, where the scope is  
 429 determined by recursively applying this algorithm.

430 **Note:** Other associations that are not listed in Table 2 may exist and may be used in Step 2 of the above  
 431 algorithm.

432 **7.1.2 CIM\_Role.CommonName**

433 The CIM\_Role.CommonName property shall be formatted using the following algorithm:

434 < OrgID > : < LocalID >, where < OrgID > and < LocalID > are separated by a colon (:), and where  
 435 < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the  
 436 business entity that is creating or defining the CommonName or that is a registered ID assigned to the  
 437 business entity by a recognized global authority. (This requirement is similar to the < Schema Name > \_  
 438 < Class Name > structure of Schema class names.) In addition, to ensure uniqueness, < OrgID > shall  
 439 not contain a colon (:). The first colon to appear in this property shall appear between < OrgID > and <  
 440 LocalID >. < LocalID > is chosen by the business entity and should not be reused to identify different  
 441 underlying (real-world) elements.

442 **7.1.3 Privileges of Authorized Role**

443 The privileges of an authorized role may be represented by instances of CIM\_Privilege. If the  
 444 CIM\_Role.RoleCharacteristics property contains the value 3 (Opaque), no instances of CIM\_Privilege  
 445 shall be associated with the instance of CIM\_Role through the CIM\_MemberOfCollection association.



446 If the CIM\_Role.RoleCharacteristics property does not contain the value 3 (Opaque), zero or more  
447 instances of CIM\_Privilege shall be associated with the instance of CIM\_Role through the  
448 CIM\_MemberOfCollection association.

449 The three types of CIM\_Privilege instances are Denied Privileges, Granted Privileges, and Template  
450 Privileges (see sections 3.18, 3.19, and 3.22).

### 451 7.1.3.1 Granted Privileges and Denied Privileges

452 Granted Privileges and Denied Privileges are associated with instances of CIM\_Role through instances of  
453 CIM\_MemberOfCollection. If at least one instance of CIM\_Privilege is associated with an instance of  
454 CIM\_Role, at least one Granted Privilege shall be associated with the instance of CIM\_Role. Any  
455 activities that are not represented by Granted Privileges associated with an instance of CIM\_Role are  
456 assumed as denied activities for the role.

457 If the instance of CIM\_Role is associated with Denied Privileges and Granted Privileges, the Denied  
458 Privileges shall take precedence over the Granted Privileges.

### 459 7.1.3.2 Cumulative Privileges for a Role

460 More than one Granted Privilege and more than one Denied Privilege can be associated with an instance  
461 of CIM\_Role. This section defines an algorithm to accumulate all the rights for a given role into one  
462 conceptual instance of CIM\_Privilege, Cumulative Role Privilege (see section 3.16). Upon completion of  
463 this algorithm, the Cumulative Role Privilege will reflect the rights explicitly granted by the instance of  
464 CIM\_Role.

465 The following algorithm shall be used to construct Cumulative Role Privilege:

- 466 1) Select all the Granted Privileges (instances of CIM\_Privilege with the PrivilegeGranted property set  
467 to TRUE) that are associated with the given CIM\_Role instance through CIM\_MemberOfCollection  
468 associations.
- 469 2) For each instance of Granted Privileges, select the CIM\_Privilege.Activities,  
470 CIM\_Privilege.ActivityQualifiers, and CIM\_Privilege.QualifierFormats array properties.
- 471 3) For each element in the CIM\_Privilege.Activities property array, select the value of the corresponding  
472 index of CIM\_Privilege.Activities, CIM\_Privilege.ActivityQualifiers, and  
473 CIM\_Privilege.QualifierFormats property arrays,
  - 474 – Determine if the Cumulative Role Privilege's CIM\_Privilege.Activities,  
475 CIM\_Privilege.ActivityQualifiers, and CIM\_Privilege.QualifierFormats property arrays contain the  
476 combination of selected element values from step 3.
  - 477 – If not, add the combination of selected values to the appropriate array properties of Cumulative  
478 Role Privilege.
- 479 4) Select all the Denied Privileges (instances of CIM\_Privilege with the PrivilegeGranted property set to  
480 FALSE) that are associated with the given CIM\_Role instance through CIM\_MemberOfCollection  
481 associations.
- 482 5) For each instance of Denied Privileges, select the CIM\_Privilege.Activities,  
483 CIM\_Privilege.ActivityQualifiers, and CIM\_Privilege.QualifierFormats array properties.
- 484 6) For each element in the CIM\_Privilege.Activities property array, select the value of the corresponding  
485 index of CIM\_Privilege.Activities, CIM\_Privilege.ActivityQualifiers, and  
486 CIM\_Privilege.QualifierFormats property arrays,
  - 487 – Determine if the Cumulative Role Privilege's CIM\_Privilege.Activities,  
488 CIM\_Privilege.ActivityQualifiers, and CIM\_Privilege.QualifierFormats property arrays contain the  
489 combination of selected element values.

- 490 – If it does, remove the combination of selected values from the appropriate array properties of  
491 Cumulative Role Privilege.

492 If the CIM\_Privilege.Activities, CIM\_Privilege.ActivityQualifiers, or CIM\_Privilege.QualifierFormats  
493 property is Null for all instances of CIM\_Privilege where the CIM\_Privilege.PrivilegeGranted property has  
494 the value TRUE, the property shall be Null for the Cumulative Role Privilege.

### 495 7.1.3.3 Cumulative Privileges for Multiple Roles

496 The Cumulative Privilege granted by the instances of CIM\_Role in an arbitrary set of instances of  
497 CIM\_Role shall be defined as follows:

- 498 1) For each instance of CIM\_Role in the set, follow the algorithm in section 7.1.3.2 to construct the  
499 Cumulative Role Privileges for the instance.
- 500 2) For each instance of Cumulative Role Privileges,  
501 – For each element in the CIM\_Privilege.Activities property array, select the value of the  
502 corresponding index of CIM\_Privilege.Activities, CIM\_Privilege.ActivityQualifiers, and  
503 CIM\_Privilege.QualifierFormats property arrays,  
504 1) Determine if the Cumulative Privilege's CIM\_Privilege.Activities,  
505 CIM\_Privilege.ActivityQualifiers, and CIM\_Privilege.QualifierFormats property arrays  
506 contain the combination of selected element values from step 1.  
507 2) If not, add the combination of selected values to the appropriate array properties of  
508 Cumulative Role Privilege.

### 509 7.1.3.4 Template Privileges

---

#### 510 EXPERIMENTAL

511 Template Privileges are used to provide the client with guidance for the Privileges parameter of the  
512 CIM\_RoleBasedAuthorizationService.CreateRole() and  
513 CIM\_RoleBasedAuthorizationService.ModifyRole() methods. An element in the array of the Privileges  
514 parameter of these methods may be created from Template Privileges by replicating all the properties of  
515 a Template Privilege with the exception of keys.

#### 516 EXPERIMENTAL

---

517 The Template Privileges shall be associated with instances of CIM\_RoleBasedAuthorizationService  
518 through instances of CIM\_ConcreteDependency.

### 519 7.1.4 Static Authorized Role

520 An authorized role that cannot be modified or deleted by the instrumentation is referred to as a static  
521 authorized role. The CIM\_Role.RoleCharacteristics property shall contain the value 2 (Static Role) for an  
522 instance of CIM\_Role that represents a static authorized role. The CIM\_Role instance that represents the  
523 static authorized role shall not support Authorized Role Management as described in section 7.2.

## 524 7.2 Authorized Role Management

525 This clause details the requirements related to managing the roles and privileges. If role and privilege  
526 management is supported, the requirements specified in this clause shall be met.

527 Authorized Role Management provides functionality for creating, deleting, and modifying instances of  
528 CIM\_Role, associated instances of CIM\_Privilege, and necessary associations.

529 Authorized Role Management shall be supported for an instance of CIM\_Role if and only if the  
530 SupportedMethods property array of the Associated Role Management Capability of the CIM\_Role  
531 instance contains at least one value, and if and only if the CIM\_Role.RoleCharacteristics property does  
532 not contain the value 2 (Static).

533 Authorized Role Management consists of support for one or more of the following functionalities:

---

#### 534 **EXPERIMENTAL**

- 535 • Creation of a CIM\_Role instance and associated CIM\_Privilege instances by using the  
536 CIM\_RoleBasedAuthorizationService.CreateRole() method. See section 8.1 for requirement details.
- 537 • Deletion of a CIM\_Role instance and associated CIM\_Privilege instances by using the  
538 CIM\_RoleBasedAuthorizationService.DeleteRole() method. See section 8.1.1 for requirement  
539 details.

#### 540 **EXPERIMENTAL**

---

- 541 • Modification of a CIM\_Role instance and associated CIM\_Privilege instances by using the  
542 CIM\_RoleBasedAuthorizationService.ModifyRole() method. See section 8.2.1 for requirement  
543 details.
- 544 • Modification of a CIM\_Privilege instance by using the ModifyInstance operation. See section 8.13 for  
545 requirement details.

### 546 **7.3 Authorized Role Membership of Security Principal**

547 The privileges for a security principal may be managed. This behavior is optional. If this behavior is  
548 implemented, the requirements specified in the following sections shall be implemented.

549 The *Simple Identity Management Profile* shall be implemented.

#### 550 **7.3.1 Roles Available to Principal**

551 For each instance of CIM\_Role with which an instance of CIM\_Identity may be associated through the  
552 CIM\_MemberOfCollection association, an instance of CIM\_ServiceServiceDependency shall associate at  
553 least one CIM\_AccountManagementService instance that is associated through the  
554 CIM\_ServiceAffectsElement association with the CIM\_Identity instance to the instance of  
555 CIM\_RoleBasedAuthorizationService that is associated through the CIM\_ServiceAffectsElement  
556 association to the instance of CIM\_Role.

#### 557 **7.3.2 Managing Privileges through Role Assignment**

558 Privileges for a principal may be managed by assigning the principal to zero or more roles. An instance of  
559 CIM\_Identity shall be a member of an instance of CIM\_Role, if and only if an instance of  
560 CIM\_MemberOfCollection associates the instance of CIM\_Identity that represents the principal with the  
561 instance of CIM\_Role that represents a role assigned to the principal.

562 If the CIM\_Identity instance is not associated with any instances of CIM\_Role through the  
563 CIM\_MemberOfCollection association, the principal shall not have any privileges.

#### 564 **7.3.3 Managing Privileges One to One for a Principal**

565 The privileges for an authenticated entity may be modeled through a one-to-one correspondence of  
566 instances of CIM\_Role with an instance of CIM\_Identity. If privileges are managed through one-to-one  
567 correspondence, the requirements specified in this section shall be met.

568 Exactly one instance of CIM\_ConcreteDependency shall be implemented as defined in section 10.2 that  
569 associates the CIM\_Identity instance with a CIM\_Role instance. At most one instance of CIM\_Identity  
570 shall be associated with the CIM\_Role instance through the CIM\_MemberOfCollection association, if the  
571 CIM\_Role instance is referenced by a CIM\_ConcreteDependency association. The instance relationship  
572 through CIM\_ConcreteDependency is used to indicate that the CIM\_Role instance can be used for the  
573 single CIM\_Identity instance with which it is associated.

## 574 **7.4 Privilege Management Capability**

575 This section provides requirements for identifying the Associated Privilege Management Capability for an  
576 instance of CIM\_Privilege. Each instance of CIM\_Privilege associated with a CIM\_Role instance through  
577 CIM\_MemberOfCollection association shall have the capabilities defined in the Associated Privilege  
578 Management Capability. CIM\_Privilege may be optionally associated with  
579 CIM\_RoleBasedAuthorizationService through CIM\_ServiceAffectsElement association.

580 If there is an instance of CIM\_ServiceAffectsElement associating the instance of CIM\_Privilege with an  
581 instance of CIM\_RoleBasedAuthorizationService, then the instance of  
582 CIM\_RoleBasedManagementCapabilities associated with the instance of  
583 CIM\_RoleBasedAuthorizationService shall be the Associated Privilege Management Capability.

584 If there is an instance of CIM\_ServiceAffectsElement associating the instance of CIM\_Privilege with an  
585 instance of CIM\_RoleBasedAuthorizationService, the Associated Role Capability of instance(s) of  
586 CIM\_Role associated with the instance of CIM\_Privilege through CIM\_MemberOfCollection association(s)  
587 shall be the Associated Privilege Management Capability.

### 588 **7.4.1.1 Shared Privileges**

589 If the CIM\_RoleBasedManagementCapabilities.SharedPrivilegeSupported property is set to FALSE, the  
590 instance of CIM\_Privilege shall be associated to only one instance of CIM\_Role.

591 If the CIM\_RoleBasedManagementCapabilities.SharedPrivilegeSupported property is set to TRUE, the  
592 instance of CIM\_Privilege may be associated to one or more instances of CIM\_Role.

### 593 **7.4.1.2 Supported Activities**

594 This clause details the requirements related to representation of the list of supported activities of the  
595 privileges. This behavior is optional. If the representation of the list of supported activities of the privileges  
596 is supported, the requirements specified in this clause shall be met.

597 The ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties of the  
598 Associated Privilege Management Capability represents the full list of supported activities of the privilege.

599 If the ModifyInstance operation is supported on an instance of CIM\_Privilege, the ActivitiesSupported,  
600 ActivityQualifiersSupported, and QualifierFormatsSupported properties on the Associated Privilege  
601 Management Capability of the instance of CIM\_Privilege shall be supported.

602 If the implementation supports the ActivitesSupported property, than the ActivityQualifiersSupported shall  
603 be implemented, and the QualifierFormats may be implemented.

604 The ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties of the  
605 Associated Privilege Management Capability of the instance of CIM\_Privilege shall represent the super  
606 set of supported activities, and the following rules apply:

- 607 • The CIM\_Privilege.Activities property array shall contain a subset of elements of the  
608 ActivitiesSupported property array elements.
- 609 • The CIM\_Privilege.ActivityQualifiers property array shall contain a subset of elements of the  
610 ActivityQualifiersSupported property array elements.

- 611 • The CIM\_Privilege.QualifierFormats property array shall contain a subset of elements of the  
612 QualifierFormatsSupported property array elements.

## 613 8 Methods

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

---

### 616 EXPERIMENTAL

#### 617 8.1 CIM\_RoleBasedAuthorizationService.CreateRole()

618 The CreateRole() method is used to create a new authorized role with specific privileges.

619 Upon the successful execution of the CreateRole() method:

- 620 • An instance of CIM\_Role shall exist that is the exact replica of the embedded instance of CIM\_Role  
621 of the RoleTemplate parameter except for the key properties.
- 622 • An instance of the CIM\_OwningCollectionElement association shall associate the new CIM\_Role  
623 instance and the scoping CIM\_ComputerSystem instance referenced by the OwningSystem  
624 parameter.
- 625 • Instances of CIM\_Privilege shall be associated with the newly created instance of CIM\_Role through  
626 the CIM\_MemberOfCollection association.
- 627 • The Cumulative Role Privilege of the newly associated instances of CIM\_Privilege shall be equal to  
628 the Cumulative Role Privilege of the embedded instances of CIM\_Privilege contained in the  
629 Privileges parameter.
- 630 • If the SharedPrivilegeSupported property of the CIM\_RoleBasedManagementCapabilities instance  
631 that is associated with the CIM\_RoleBasedAuthorizationService instance has a value of FALSE, the  
632 CIM\_Privilege instances shall be associated only with the newly created CIM\_Role instance and  
633 shall not be associated with any other instance of CIM\_Role.
- 634 • If the SharedPrivilegeSupported property of the CIM\_RoleBasedManagementCapabilities instance  
635 that is associated with the CIM\_RoleBasedAuthorizationService instance has a value of TRUE, the  
636 CIM\_Privilege instances shall be associated with the newly created CIM\_Role instance and may be  
637 associated with any other instance of CIM\_Role.
- 638 • Instances of CIM\_RoleLimitedToTarget shall associate the newly created CIM\_Role instance with  
639 the instances referenced by the RoleLimitedToTargets parameter.
- 640 • Instances of CIM\_ServiceAffectsElement shall associate the new CIM\_Role instance and the  
641 CIM\_RoleBasedAuthorizationService instance.

642 If the properties of the embedded instances of RoleTemplate parameters and privileges are not fully  
643 specified, the implementation may use its defaults to populate the resulting instances of CIM\_Role and  
644 CIM\_Privilege.

645 The CreateRole() method shall return the value 2 (Error occurred) if the RoleCharacteristics property of  
646 the RoleTemplate parameter's instance of CIM\_Role contains the value 2 (Static).

647 The CreateRole() method's return code values shall be as specified in Table 3 where the method  
648 execution behavior matches the return code description. The CreateRole() method's parameters are  
649 specified in Table 4.

650 No standard messages are defined for this method.

651 **Table 3 – CIM\_RoleBasedAuthorizationService.CreateRole() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.

652 **Table 4 – CIM\_RoleBasedAuthorizationService.CreateRole() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	RoleTemplate	string	Embedded instance of CIM_Role that contains the non-key properties for the desired CIM_Role instance
IN	OwningSystem	CIM_ComputerSystem REF	References the CIM_ComputerSystem to which the new CIM_Role instance is going to be scoped
IN, REQ	Privileges	string []	Array of embedded instances of CIM_Privilege that describe the instances of CIM_Privilege to be associated with the desired CIM_Role instance
IN	RoleLimitedToTargets	CIM_ManagedElement REF []	References to the instances of CIM_ManagedElement subclasses to which the desired CIM_Role instance will be constrained
OUT	Role	CIM_Role REF	Reference to the desired newly created CIM_Role instance

653 **8.1.1 CIM\_RoleBasedAuthorizationService.CreateRole() Conditional Support**

654 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
 655 Role Management Capability contains the value 4 (CreateRole), the CreateRole() method shall be  
 656 implemented and shall not return the value 1 (Not Supported).

657 If Authorized Role Management is not supported or the SupportedMethods property array of the  
 658 Associated Role Management Capability does not contain the value 4 (CreateRole), the CreateRole()  
 659 method shall not be implemented or shall always return the value 1 (Not Supported).

660 **8.2 CIM\_RoleBasedAuthorizationService.DeleteRole()**

661 If the DeleteRole() method is implemented, the requirements specified in this section shall be met.

662 The execution of the DeleteRole() method shall attempt to delete the CIM\_Role instance referenced by  
 663 the Role parameter and the associated instances as described in this section.

664 If the CIM\_Role instance referenced by the Role parameter is not associated with the  
 665 CIM\_RoleBasedAuthorizationService instance through the CIM\_ServiceAffectsElement association, the  
 666 DeleteRole() method shall fail and return the value 2 (Error occurred).

667 If the DeleteRole() method is implemented and the RoleCharacteristics property of the CIM\_Role  
 668 instance referenced by the Role parameter contains a value of 2 (Static), the DeleteRole() method shall  
 669 fail and return the value 2 (Error occurred).

670 Upon the successful execution of the DeleteRole() method, the following actions occur:

- 671 • All instances of the CIM\_RoleLimitedToTarget association that reference the CIM\_Role instance that  
672 is referenced by the Role parameter shall be deleted.
- 673 • If the SharedPrivilegeSupported property of the CIM\_RoleBasedManagementCapabilities instance  
674 that is associated with the CIM\_RoleBasedAuthorizationService instance has a value of FALSE, the  
675 implementation shall delete all the CIM\_Privilege instances that are associated with the CIM\_Role  
676 instance that is referenced by the Role parameter.
- 677 • If the SharedPrivilegeSupported property of the CIM\_RoleBasedManagementCapabilities instance  
678 that is associated with the CIM\_RoleBasedAuthorizationService instance has a value of TRUE, the  
679 implementation shall delete the CIM\_Privilege instances that are only associated with the CIM\_Role  
680 instance that is referenced by the Role parameter.
- 681 • All instances of the CIM\_MemberOfCollection association that reference the CIM\_Role instance that  
682 is referenced by the Role parameter shall be deleted.
- 683 • All instances of the CIM\_OwningCollectionElement association that reference the CIM\_Role instance  
684 that is referenced by the Role parameter shall be deleted.
- 685 • The instance of the CIM\_ServiceAffectsElement association that references the CIM\_Role instance  
686 that is referenced by the Role parameter and that references the  
687 CIM\_RoleBasedAuthorizationService instance shall be deleted.

688 The DeleteRole() method's return code values shall be as specified in Table 5 where the method  
689 execution behavior matches the return code description. The DeleteRole() method's parameters are  
690 specified in Table 6.

691 No standard messages are defined for this method.

692 **Table 5 – CIM\_RoleBasedAuthorizationService.DeleteRole() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.

693 **Table 6 – CIM\_RoleBasedAuthorizationService.DeleteRole() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	Role	CIM_Role REF	The reference to the CIM_Role instance to be deleted

694 **8.2.1 CIM\_RoleBasedAuthorizationService.DeleteRole() Conditional Support**

695 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
696 Role Management Capability contains the value 9 (DeleteRole), the DeleteRole() method shall be  
697 implemented and shall not return the value 1 (Not Supported).

698 If Authorized Role Management is not supported or the SupportedMethods property array of the  
699 Associated Role Management Capability does not contain the value 9 (DeleteRole), the DeleteRole()  
700 method shall not be implemented or shall always return the value 1 (Not Supported).

701 **EXPERIMENTAL**

---

702 **8.3 CIM\_RoleBasedAuthorizationService.ModifyRole()**

703 The ModifyRole() method is used to modify an authorized role and its privileges.

704 Upon the successful execution of the ModifyRole() method, the following actions occur:

- 705 • If the Privileges parameter is Null, the instances of CIM\_Privilege that are associated with the  
706 Modified Role shall not be modified (see section 3.20).
- 707 • If the Privileges parameter is not Null and instances of CIM\_Privilege are associated with the  
708 Modified Role through the CIM\_MemberOfCollection association, the Cumulative Role Privilege of  
709 the associated instances of CIM\_Privilege shall be equal to the Cumulative Role Privilege of the  
710 embedded instances of CIM\_Privilege that are contained in the Privileges parameter.
- 711 • If the SharedPrivilegeSupported property of the CIM\_RoleBasedManagementCapabilities instance  
712 that is associated with the CIM\_RoleBasedAuthorizationService instance has a value of FALSE, the  
713 CIM\_Privilege instances shall be associated only with the Modified Role and shall not be associated  
714 with any other instance of CIM\_Role.
- 715 • If the SharedPrivilegeSupported property of the CIM\_RoleBasedManagementCapabilities instance  
716 that is associated with the CIM\_RoleBasedAuthorizationService instance has a value of TRUE, the  
717 CIM\_Privilege instances shall be associated with the Modified Role and may be associated with any  
718 other instance of CIM\_Role.
- 719 • An instance of CIM\_RoleLimitedToTarget shall reference the Modified Role and an instance of  
720 CIM\_ManagedElement only if a reference to the CIM\_ManagedElement was contained in the  
721 RoleLimitedToTargets parameter.

722 The ModifyRole() method shall return the value 2 (Error occurred) if the Modified Role is not associated  
723 with the instance of CIM\_RoleBasedAuthorizationService through an instance of  
724 CIM\_ServiceAffectsElement.

725 The ModifyRole() method shall return the value 2 (Error occurred) if the Modified Role  
726 RoleCharacteristics property contains the value 2 (Static).

727 The ModifyRole() method's return code values shall be as specified in Table 7 where the method  
728 execution behavior matches the return code description. The ModifyRole() method's parameters are  
729 specified in Table 8.

730 No standard messages are defined for this method.

731 **Table 7 – CIM\_RoleBasedAuthorizationService.ModifyRole() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is not supported in the implementation.
2	Error occurred.

732 **Table 8 – CIM\_RoleBasedAuthorizationService.ModifyRole() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Privileges	string []	Array of embedded instances of CIM_Privilege that describe the complete set of instances of CIM_Privilege to be associated with the Modified Role
IN	RoleLimitedToTargets	CIM_ManagedElement REF []	References to the instances of CIM_ManagedElement subclasses to which the Modified Role will be constrained
IN, REQ	Role	CIM_Role REF	Reference to Modified Role



733 **8.3.1 CIM\_RoleBasedAuthorizationService.ModifyRole() Conditional Support**

734 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
 735 Role Management Capability contains the value 5 (ModifyRole), the ModifyRole() method shall be  
 736 implemented and shall not return the value 1 (Not Supported).

737 If Authorized Role Management is not supported or the SupportedMethods property array of the  
 738 Associated Role Management Capability does not contain the value 5 (ModifyRole), the ModifyRole()  
 739 method shall not be implemented or shall always return the value 1 (Not Supported).

740 **8.4 CIM\_RoleBasedAuthorizationService.AssignRoles()**

741 The AssignRoles() method is used to assign a security principal that is represented by an instance of  
 742 CIM\_Identity to zero or more roles represented by instances of CIM\_Role.

743 If the CIM\_Identity instance identified by the Identity parameter is not associated with an instance of  
 744 CIM\_AccountManagementService through the CIM\_ServiceAffectsElement association, where the  
 745 CIM\_AccountManagementService is associated through the CIM\_ServiceServiceDependency  
 746 association with the instance of CIM\_RoleBasedAuthorizationService upon which the method was  
 747 invoked, the method shall return the value 2 (Failed).

748 If the Roles parameter contains a reference to an instance of CIM\_Role that is not associated through the  
 749 CIM\_ServiceAffectsElement association with the instance of CIM\_RoleBasedAuthorizationService upon  
 750 which the method was invoked, the method shall return the value 2 (Failed).

751 The AssignRoles() method's return code values shall be as specified in Table 9 where the method  
 752 execution behavior matches the return code description. The AssignRoles() method's parameters are  
 753 specified in Table 10.

754 No standard messages are defined for this method.

755 **Table 9 – CIM\_RoleBasedAuthorizationService.AssignRoles() Method: Return Code Values**

Value	Description
0	Operation completed successfully.
1	Operation unsupported
2	Failed

756 **Table 10 – CIM\_RoleBasedAuthorizationService.AssignRoles() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	Identity	CIM_Identity REF	Reference to the CIM_Identity instance that represents the security principal
IN, REQ	Roles	CIM_Role[] REF	Array of references to instances of CIM_Role

757 **8.4.1 CIM\_RoleBasedAuthorizationService.AssignRoles() Conditional Support**

758 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
 759 Role Management Capability contains the value 6 (AssignRoles), the AssignRoles() method shall be  
 760 implemented and shall not return the value 1 (Not Supported).

761 If Authorized Role Management is not supported or the SupportedMethods property array of the  
 762 Associated Role Management Capability does not contain the value 6 (AssignRoles), the AssignRoles()  
 763 method shall not be implemented or shall always return the value 1 (Not Supported).

764 **8.5 CIM\_RoleBasedAuthorizationService.ShowAccess()**

765 The ShowAccess() method is used to query the rights granted to a security principal for a managed  
766 element.

767 If the Subject or Target parameter is Null, the method shall return the value 2 (Failed).

768 If the Subject parameter is not an instance of CIM\_Identity, the method shall return the value 2 (Failed).

769 If the CIM\_Identity instance identified by the Subject parameter is not associated with an instance of  
770 CIM\_AccountManagementService instance through the CIM\_ServiceAffectsElement association, where  
771 the CIM\_AccountManagementService is associated through the CIM\_ServiceServiceDependency  
772 association with the instance of CIM\_RoleBasedAuthorizationService upon which the method was  
773 invoked, the method shall return the value 2 (Failed).

774 Upon successful completion, the method shall return the value 0 and the Privileges Out parameter shall  
775 be the Cumulative Privilege defined in section 7.1.3.3, where

- 776 • the set of instances of CIM\_Role are those instances such that the instance of CIM\_Identity specified  
777 by the Subject parameter is a member of the CIM\_Role instance as defined in section 7.3.2
- 778 • the instance of CIM\_ManagedElement specified by the Target parameter is in the scope of the  
779 CIM\_Role instance as defined in section 7.1.1.1
- 780 • the instance of CIM\_Role is associated with the instance of CIM\_RoleBasedAuthorizationService  
781 through the CIM\_ServiceAffectsElement association

782 The OutSubjects and OutTargets parameters shall be Null if the method completes.

783 The ShowAccess() method's return code values shall be as specified in Table 11 where the method  
784 execution behavior matches the return code description. The ShowAccess() method's parameters are  
785 specified in Table 12.

786 No standard messages are defined for this method.

787 **Table 11 – CIM\_RoleBasedAuthorizationService.ShowAccess() Method: Return Code Values**

Value	Description
0	Operation completed successfully.
1	Operation unsupported.
2	Failed

788 **Table 12 – CIM\_RoleBasedAuthorizationService.ShowAccess() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Subject	CIM_ManagedElement REF	Reference to the CIM_Identity instance that represents the security principal
IN	Target	CIM_ManagedElement REF	Reference to the CIM_ManagedElement instance that represents the target
OUT	Privileges[]	string	Array that contains the embedded instances of the Cumulative Privilege
OUT	OutSubjects[]	CIM_ManagedElement REF	This output parameter shall be always NULL.
OUT	OutTargets[]	CIM_ManagedElement REF	This output parameter shall be always NULL.

### 789 **8.5.1 CIM\_RoleBasedAuthorizationService.ShowAccess() Conditional Support**

790 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
791 Role Management Capability contains the value 1 (ShowAccess), the ShowAccess() method shall be  
792 implemented and shall not return the value 1 (Not Supported).

793 If Authorized Role Management is not supported or the SupportedMethods property array of the  
794 Associated Role Management Capability does not contain the value 1 (ShowAccess), the ShowAccess()  
795 method shall not be implemented or shall always return the value 1 (Not Supported).

### 796 **8.6 CIM\_RoleBasedAuthorizationService.ShowRoles()**

797 The ShowRoles() method is used to show the roles that the specified security principal is a member of  
798 and the specified managed element is within the scope of.

799 If the Subject parameter is not an instance of CIM\_Identity, the method shall return the value 2 (Failed).

800 If the Subject parameter is not Null and the CIM\_Identity instance identified by the Subject parameter is  
801 not associated with an instance of CIM\_AccountManagementService through the  
802 CIM\_ServiceAffectsElement association, where the CIM\_AccountManagementService is associated  
803 through the CIM\_ServiceServiceDependency association with the instance of  
804 CIM\_RoleBasedAuthorizationService upon which the method was invoked, the method shall return the  
805 value 2 (Failed).

806 Upon successful completion, the method shall return the value 0.

807 If the Subject and Target parameters are not Null, upon successful completion of the method

- 808 • the Roles parameter shall contain an embedded instance of CIM\_Role for each instance of  
809 CIM\_Role such that the instance of CIM\_Identity specified by the Subject parameter is a member of  
810 the CIM\_Role instance as defined in section 7.3.2
- 811 • the instance of CIM\_ManagedElement specified by the Target parameter is in the scope of the  
812 CIM\_Role instance as defined in section 7.1.1.1
- 813 • the instance of CIM\_Role is associated with the instance of CIM\_RoleBasedAuthorizationService  
814 through the CIM\_ServiceAffectsElement association

815 If the Subject parameter is not Null and the Target parameter is Null, upon successful completion of the  
816 method

- 817 • the Roles parameter shall contain an embedded instance of CIM\_Role for each of instance of  
818 CIM\_Role such that the instance of CIM\_Identity specified by the Subject parameter is a member of  
819 the CIM\_Role instance as defined in section 7.3.2
- 820 • the instance of CIM\_Role is associated with the instance of CIM\_RoleBasedAuthorizationService  
821 through the CIM\_ServiceAffectsElement association

822 If the Subject parameter is Null and the Target parameter is not Null, upon successful completion of the  
823 method

- 824 • the Roles parameter shall contain an embedded instance of CIM\_Role for each of instance of  
825 CIM\_Role such that the instance of CIM\_ManagedElement specified by the Target parameter is in  
826 the scope of the CIM\_Role instance as defined in section 7.1.1.1
- 827 • the instance of CIM\_Role is associated with the instance of CIM\_RoleBasedAuthorizationService  
828 through the CIM\_ServiceAffectsElement association

829 If the Subject and Target parameters are both Null, upon successful completion of the method, the Roles  
 830 parameter shall contain an embedded instance of CIM\_Role for each of instance of CIM\_Role such that  
 831 the instance of CIM\_Role is associated with the instance of CIM\_RoleBasedAuthorizationService through  
 832 the CIM\_ServiceAffectsElement association.

833 For each instance of CIM\_Role for which the Roles parameter contains an embedded instance of  
 834 CIM\_Role, the Privileges parameter shall contain at the same array index an embedded instance of  
 835 CIM\_Privilege that represents the Cumulative Privilege of the CIM\_Role as defined in section 7.1.3.2.

836 The ShowRoles() method's return code values shall be as specified in Table 13 where the method  
 837 execution behavior matches the return code description. The ShowRoles() method's parameters are  
 838 specified in Table 14.

839 No standard messages are defined for this method.

840 **Table 13 – CIM\_RoleBasedAuthorizationService.ShowRoles() Method: Return Code Values**

Value	Description
0	Operation completed successfully.
1	Operation unsupported.
2	Failed

841 **Table 14 – CIM\_RoleBasedAuthorizationService.ShowRoles() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Subject	CIM_Identity REF	Reference to the CIM_Identity instance that represents the security principal
IN	Target	CIM_ManagedElement REF	Reference to the CIM_ManagedElement instance
OUT	Roles[]	string	Array of embedded instances of CIM_Role
OUT	Privileges[]	string	Array of embedded instances of CIM_Privilege

842 **8.6.1 CIM\_RoleBasedAuthorizationService.ShowRoles() Conditional Support**

843 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
 844 Role Management Capability contains the value 7 (ShowRoles), the ShowRoles() method shall be  
 845 implemented and shall not return the value 1 (Not Supported).

846 If Authorized Role Management is not supported or the SupportedMethods property array of the  
 847 Associated Role Management Capability does not contain the value 7 (ShowRoles), the ShowRoles()  
 848 method shall not be implemented or shall always return the value 1 (Not Supported).

849 **8.7 Profile Conventions for Operations**

850 Support for operations for each profile class (including associations) is specified in the following  
 851 subclauses. Each subclause includes either the statement "All operations in the default list in section 8.7  
 852 are supported as described by [DSP0200 version 1.2](#)" or a table listing all of the operations that are not  
 853 supported by this profile or where the profile requires behavior other than that described by  
 854 [DSP0200 version 1.2](#).

855 The default list of operations is as follows:

- 856 • GetInstance
- 857 • EnumerateInstances
- 858 • EnumerateInstanceNames
- 859 • Associators
- 860 • AssociatorNames
- 861 • References
- 862 • ReferenceNames

863 A compliant implementation shall support all of the operations in the default list for each class, unless the  
 864 "Requirement" column states something other than *Mandatory*.

865 **8.8 CIM\_ConcreteDependency**

866 Table 15 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 867 or shall not be supported.

868 **Table 15 – Operations: CIM\_ConcreteDependency**

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

869 **8.9 CIM\_ElementCapabilities**

870 Table 16 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 871 or shall not be supported.

872 **Table 16 – Operations: CIM\_ElementCapabilities**

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

873 **8.10 CIM\_HostedService**

874 Table 17 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 875 or shall not be supported.

876 **Table 17 – Operations: CIM\_HostedService**

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

877 **8.11 CIM\_MemberOfCollection**

878 Table 18 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 879 or shall not be supported.

880 **Table 18 – Operations: CIM\_MemberOfCollection**

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

881 **8.12 CIM\_OwningCollectionElement**

882 Table 19 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 883 or shall not be supported.

884 **Table 19 – Operations: CIM\_OwningCollectionElement**

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

885 **8.13 CIM\_Privilege**

886 Table 20 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 887 or shall not be supported.

888 **Table 20 – Operations: CIM\_Privilege**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.13.1.	None

889 **8.13.1 CIM\_Privilege—ModifyInstance**

890 If Authorized Role Management is not supported or the SupportedMethods property array of the  
 891 Associated Privilege Management Capability of the instance of CIM\_Privilege does not contain the value  
 892 8 (ModifyPrivilege), then the ModifyInstance operation shall not be supported.

893 If Authorized Role Management is supported and the SupportedMethods property array of the Associated  
 894 Privilege Management Capability of the instance of CIM\_Privilege contains the value 8 (ModifyPrivilege),  
 895 the ModifyInstance operation shall be supported except as follows:

- 896 • The ModifyInstance operation shall not be supported on the Granted Privileges or Denied Privileges  
 897 that are associated with an instance of CIM\_Role if the CIM\_Role.RoleCharacteristics property  
 898 contains the value 2 (Static).
- 899 • The ModifyInstance operation shall not be supported on the Template Privileges.

900 **8.14 CIM\_RoleBasedManagementCapabilities**

901 All operations in the default list in section 8.7 are supported as described by [DSP0200 version 1.2](#).

902 **8.15 CIM\_Role**

903 All operations in the default list in section 8.7 are supported as described by [DSP0200 version 1.2](#).

904 **8.16 CIM\_RoleBasedAuthorizationService**

905 All operations in the default list in section 8.7 are supported as described by [DSP0200 version 1.2](#).

906 **8.17 CIM\_RoleLimitedToTarget**

907 Table 21 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 908 or shall not be supported.

909 **Table 21 – Operations: CIM\_RoleLimitedToTarget**

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

910 **8.18 CIM\_ServiceAffectsElement**

911 Table 22 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 912 or shall not be supported.

913 **Table 22 – Operations: CIM\_ServiceAffectsElement**

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

914 **8.19 CIM\_ServiceServiceDependency**

915 Table 23 lists operations that either have special requirements beyond those from [DSP0200 version 1.2](#)  
 916 or shall not be supported.

917 **Table 23 – Operations: CIM\_ServiceServiceDependency**

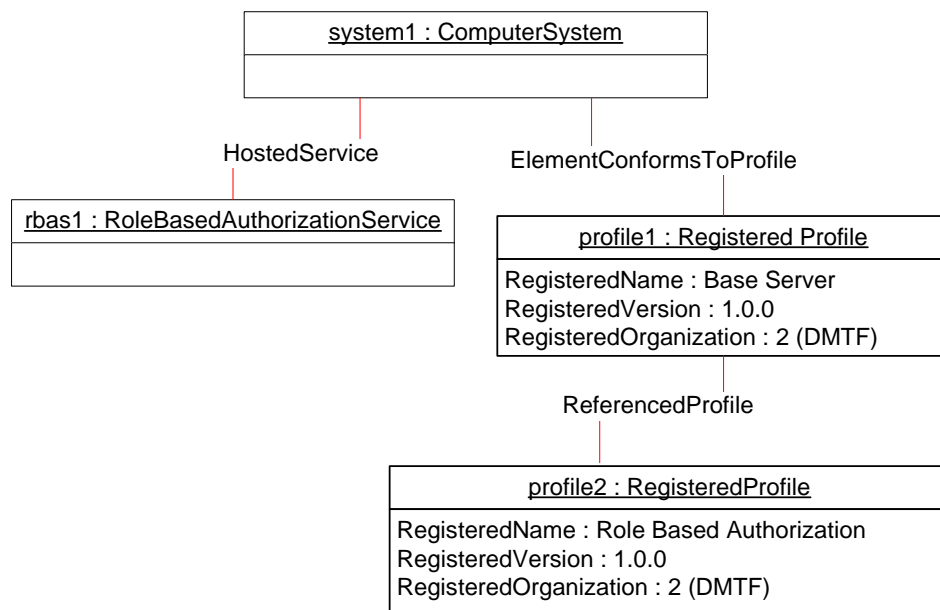
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

918 **9 Use Cases**

919 This section contains object diagrams and use cases for the *Role Based Authorization Profile*. The  
 920 contents of this section are for informative purposes only and do not constitute normative requirements  
 921 for implementations of this specification.

922 **9.1 Profile Registration**

923 Figure 2 describes one of the ways that the implementation can advertise the instantiation of the *Role*  
 924 *Based Authorization Profile*. Using scoping instance methodology as described in the *Profile Registration*  
 925 *Profile*, profile2 contains the version information for the *Role Based Authorization Profile* implementation.



926

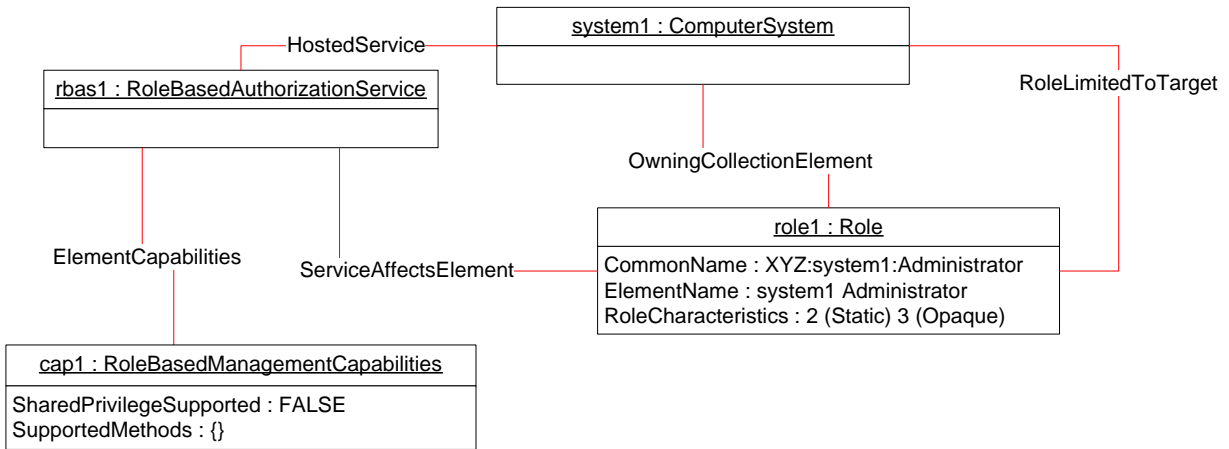
927

**Figure 2 – Profile Registration**



928 **9.2 Minimal Instantiation of the Profile**

929 Figure 3 describes a possible minimal instantiation of the *Role Based Authorization Profile*. In this  
 930 instantiation, role1 is described as being a system1 administrator role. The scope of role1 is limited to  
 931 system1 as shown by the instance of the CIM\_RoleLimitedToTarget association. role1 is opaque and  
 932 static. The rights granted by the role are not explicitly modeled. No methods are supported for  
 933 management of the role, which is indicated by the empty array for the SupportedMethods property of  
 934 cap1.

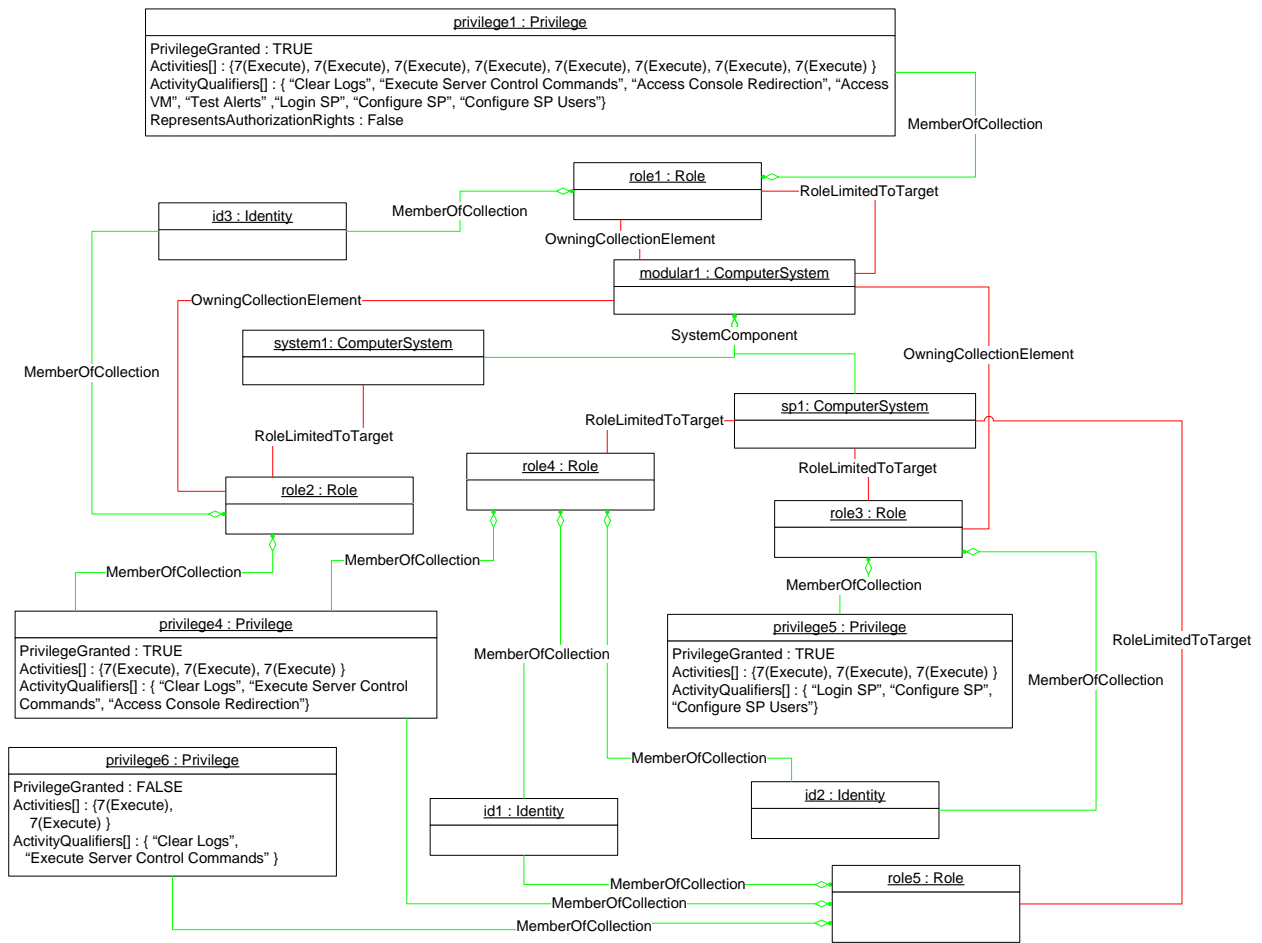


935

936 **Figure 3 – Minimal Instantiation**

937 **9.3 Evaluating Scope and Privileges**

938 Figure 4 illustrates the behavior of the CIM\_RoleBasedManagementService.ShowAccess() and  
 939 CIM\_RoleBasedManagementService.ShowRoles() methods. The diagram illustrates two systems  
 940 (system1 and sp1) contained within a third system (modular1). role1 is explicitly scoped to modular1;  
 941 system1 and sp1 are within modular1, so they are also within the scope of role1. role2 is explicitly scoped  
 942 to system1. role3, role4, and role5 are explicitly scoped to sp1.



943

944

Figure 4 – Cumulative Role Privilege Example

945 **9.3.1 CIM\_RoleBasedManagementService.ShowRoles()**

946 Given a value of id1 for the Subject parameter and Null for the Target parameter, the ShowRoles()  
 947 method will return information about each instance of CIM\_Role of which id1 is a member. Thus two  
 948 embedded instances of CIM\_Role will be in the Roles parameter, one corresponding to role5 and one  
 949 corresponding to role4. Two embedded instances of CIM\_Privilege will be returned in the Privileges  
 950 parameter, one reflecting the cumulative privileges of role5 and the other those of role4.

951 The embedded instance of CIM\_Privilege that corresponds to the Cumulative Privilege of role5  
 952 is constructed by adding the Granted Privileges (privilege4) to the Cumulative Privilege and subtracting from  
 953 the Cumulative Privilege the intersection with the Denied Privilege (privilege6). This results in the  
 954 following values for the Activities and ActivityQualifier properties:

- 955 • CIM\_Privilege.Activities = { 7(Execute) }
- 956 • CIM\_Privilege.ActivityQualifiers = { "Access Console Redirection" }

### 957 **9.3.2 CIM\_RoleBasedManagementService.ShowAccess()**

958 Each of the following sections lists a value for each of the input parameters of the ShowAccess() method  
959 and the properties of the output Privilege parameter that results from successful invocation of the method.

#### 960 **9.3.2.1 Example: CIM\_RoleBasedManagementService.ShowAccess()**

961 Subject = id1

962 Target = sp1

963 CIM\_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute) }

964 CIM\_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console  
965 Redirection" }

966 id1 belongs to role5 and role4. sp1 is in the scope of role5 and role4. The intersection of the roles is role5  
967 and role4. Therefore, the set of roles to which the algorithm in section 7.1.3.3 will be applied consists of  
968 role5 and role4. The Privileges out parameter contains the Cumulative Privilege that results from  
969 combining the Cumulative Privilege of role5 with the Cumulative Privilege of role4.

#### 970 **9.3.2.2 Example: CIM\_RoleBasedManagementService.ShowAccess()**

971 Subject = id3

972 Target = modular1

973 CIM\_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),  
974 7(Execute), 7(Execute) }

975 CIM\_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console  
976 Redirection", "Access VM", "Test Alerts", "Login SP", "Configure SP", "Configure SP Users" }

977 id3 belongs to role1 and role2. modular1 is in the scope of role1. The intersection of the roles is role1.  
978 Therefore, the set of roles to which the algorithm in section 7.1.3.3 will be applied consists of role1. The  
979 Privileges out parameter contains the Cumulative Privilege of role1.

#### 980 **9.3.2.3 Example: CIM\_RoleBasedManagementService.ShowAccess()**

981 Subject = id3

982 Target = system1

983 CIM\_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),  
984 7(Execute), 7(Execute) }

985 CIM\_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console  
986 Redirection", "Access VM", "Test Alerts", "Login SP", "Configure SP", "Configure SP Users" }

987 id3 belongs to role1 and role2. system1 is contained in modular1 and modular1 is in the scope of role1.  
988 Therefore, sp1 is in the scope of role1. system1 is explicitly within the scope of role2. The intersection of  
989 the roles is role1 and role2. Therefore, the set of roles to which the algorithm in section 7.1.3.3 will be  
990 applied consists of role1 and role2. The Cumulative Privilege of role1 is a superset of the Cumulative  
991 Privilege of role2. Therefore, the out parameter contains the Cumulative Privilege of role1.

**992 9.3.2.4 Example: CIM\_RoleBasedManagementService.ShowAccess()**

993 Subject = id3

994 Target = sp1

995 CIM\_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),  
996 7(Execute), 7(Execute) }

997 CIM\_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console  
998 Redirection", "Access VM", "Test Alerts", "Login SP", "Configure SP", "Configure SP Users" }

999 id3 belongs to role1 and role2. sp1 is contained in modular1 and modular1 is in the scope of role1.  
1000 Therefore, sp1 is in the scope of role1. The intersection of the roles is role1. Therefore, the set of roles to  
1001 which the algorithm in section 7.1.3.3 will be applied consists of role1. The Privileges out parameter  
1002 contains the Cumulative Privilege of role1.

**1003 9.3.2.5 Example: CIM\_RoleBasedManagementService.ShowAccess()**

1004 Subject = id2

1005 Target = sp1

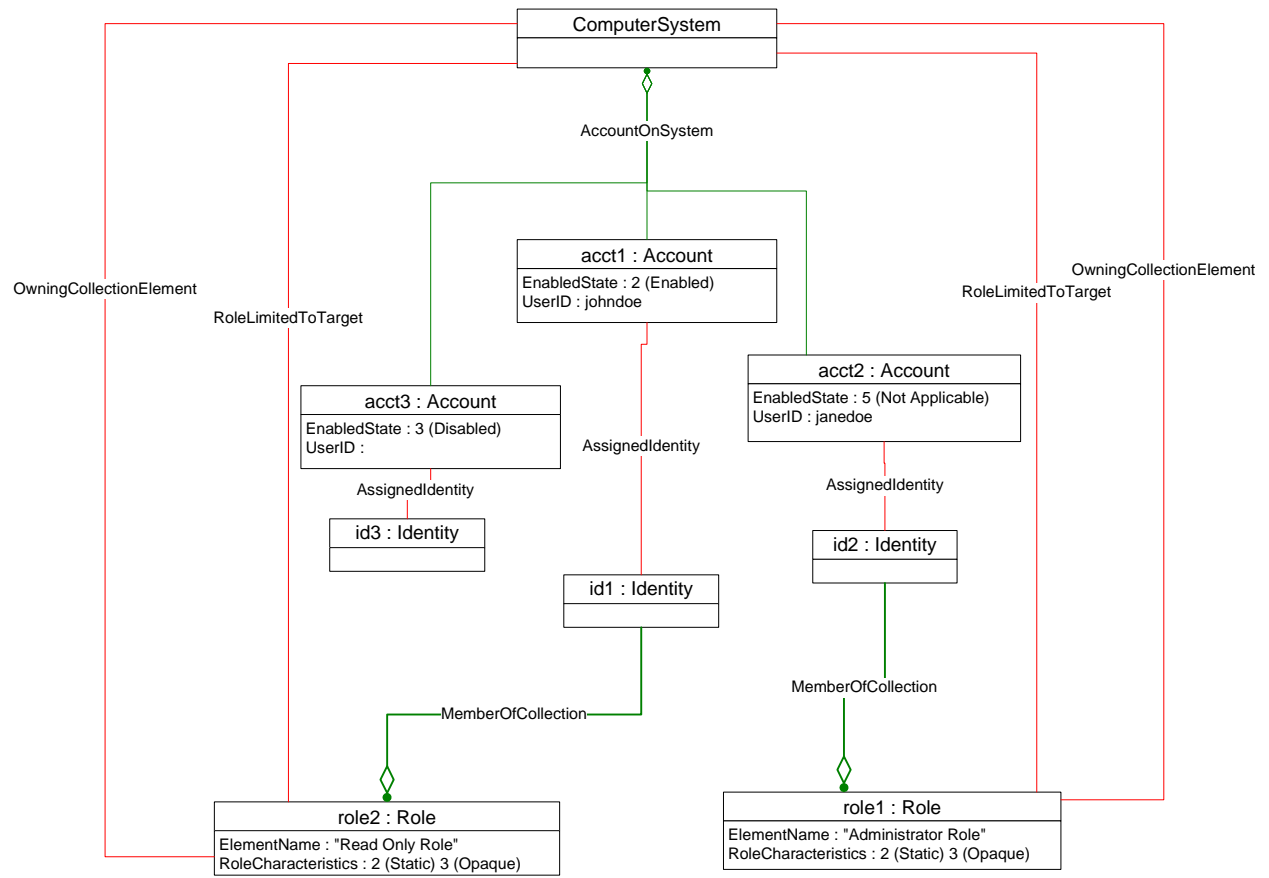
1006 CIM\_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute) }

1007 CIM\_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console  
1008 Redirection", "Login SP", "Configure SP", "Configure SP Users" }

1009 id2 belongs to role3 and role4. sp1 is in the scope of role3 and role4. The intersection of the roles is role3  
1010 and role4. Therefore, the set of roles to which the algorithm in section 7.1.3.3 will be applied consists of  
1011 role3 and role4. The Privileges out parameter contains the Cumulative Privilege that results from  
1012 combining the Cumulative Privilege of role3 with the Cumulative Privilege of role4.

**1013 9.4 Scope of the Role and Privileges for a Managed Element**

1014 Figure 5 shows a system that has three local accounts and uses role membership to manage the  
1015 privileges for a user account. This system has three local accounts: acct1, acct2, and acct3. acct1  
1016 currently has privileges of role1, and acct2 currently has the privileges of role2. acct3 does not have any  
1017 privileges. Both role1 and role2 are opaque roles based on the RoleCharacteristics property containing  
1018 value 3(Opaque), which means that their privileges are not represented by instances of CIM\_Privilege. In  
1019 this case the client is expected to know the privileges of the role by the information provided within the  
1020 CIM\_Role instance. All the CIM\_Role instances are scoped to the instance of CIM\_ComputerSystem,  
1021 which means that all the managed elements within the scope of the instance of CIM\_ComputerSystem  
1022 are within the scope of the CIM\_Role instances and the privileges of these roles are applicable on those  
1023 managed elements.

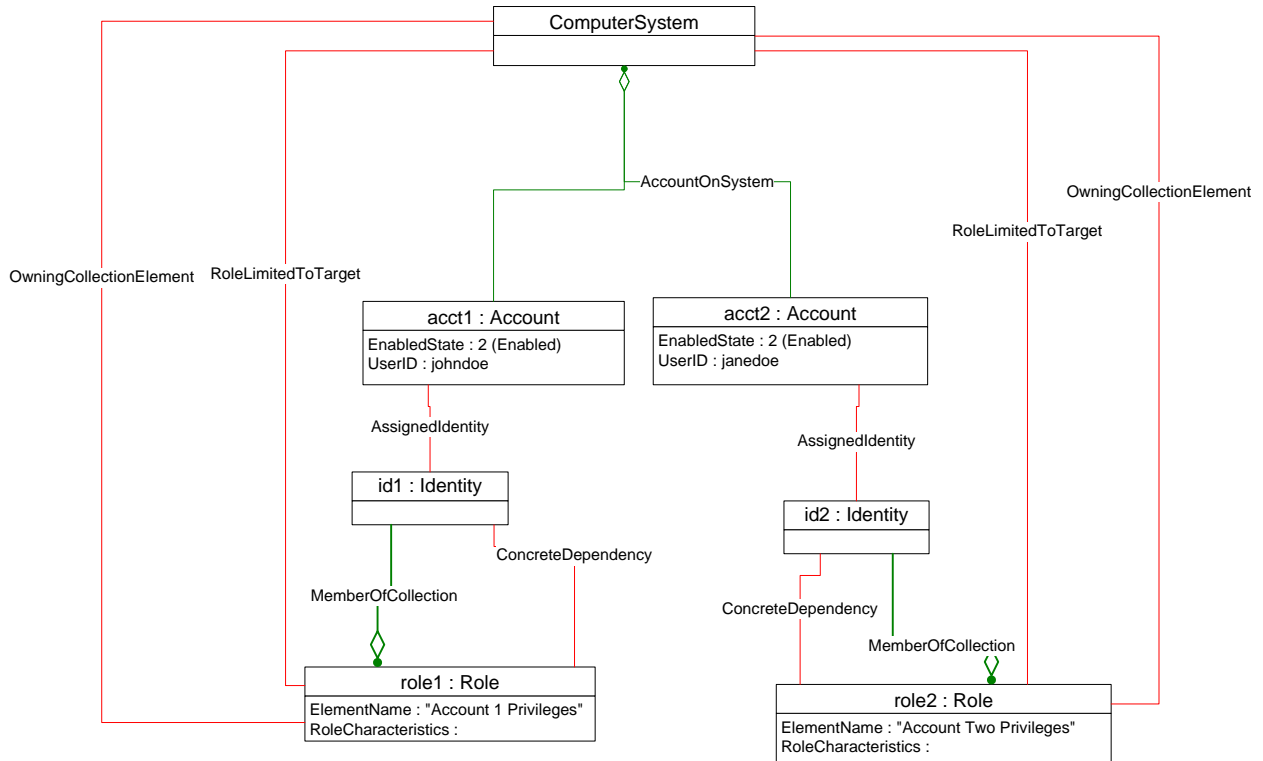


1024

1025

Figure 5 – Scope of the Roles

1026 Figure 6 shows a system that has two local accounts and manages privileges for individual accounts.  
 1027 This system has two local accounts: acct1 and acct2. Privileges for acct1 and acct2 are managed through  
 1028 role1 and role2, respectively, as indicated by the CIM\_ConcreteDependency associations. No common  
 1029 roles are defined; therefore, privileges for each account can be managed only through their respective  
 1030 dedicated roles.

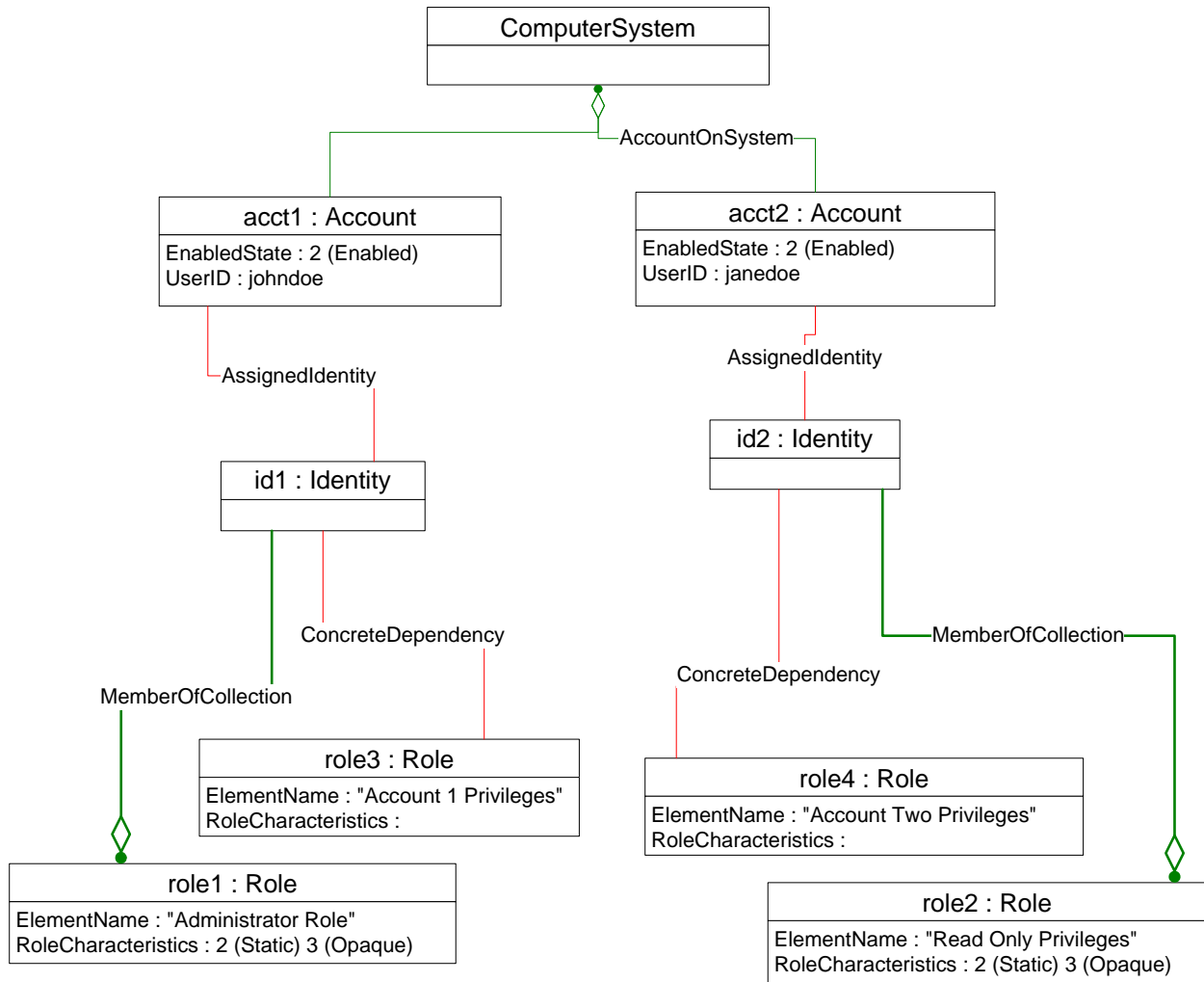


1031

1032

**Figure 6 – Fixed Accounts with Role Membership Privilege Management**

1033 Figure 7 shows a system that has two local accounts. Privileges for the accounts are managed either  
 1034 through assignment to a pre-defined role (role1 and role2) or through modification of privileges granted to  
 1035 a dedicated role (role3 and role4).



1036

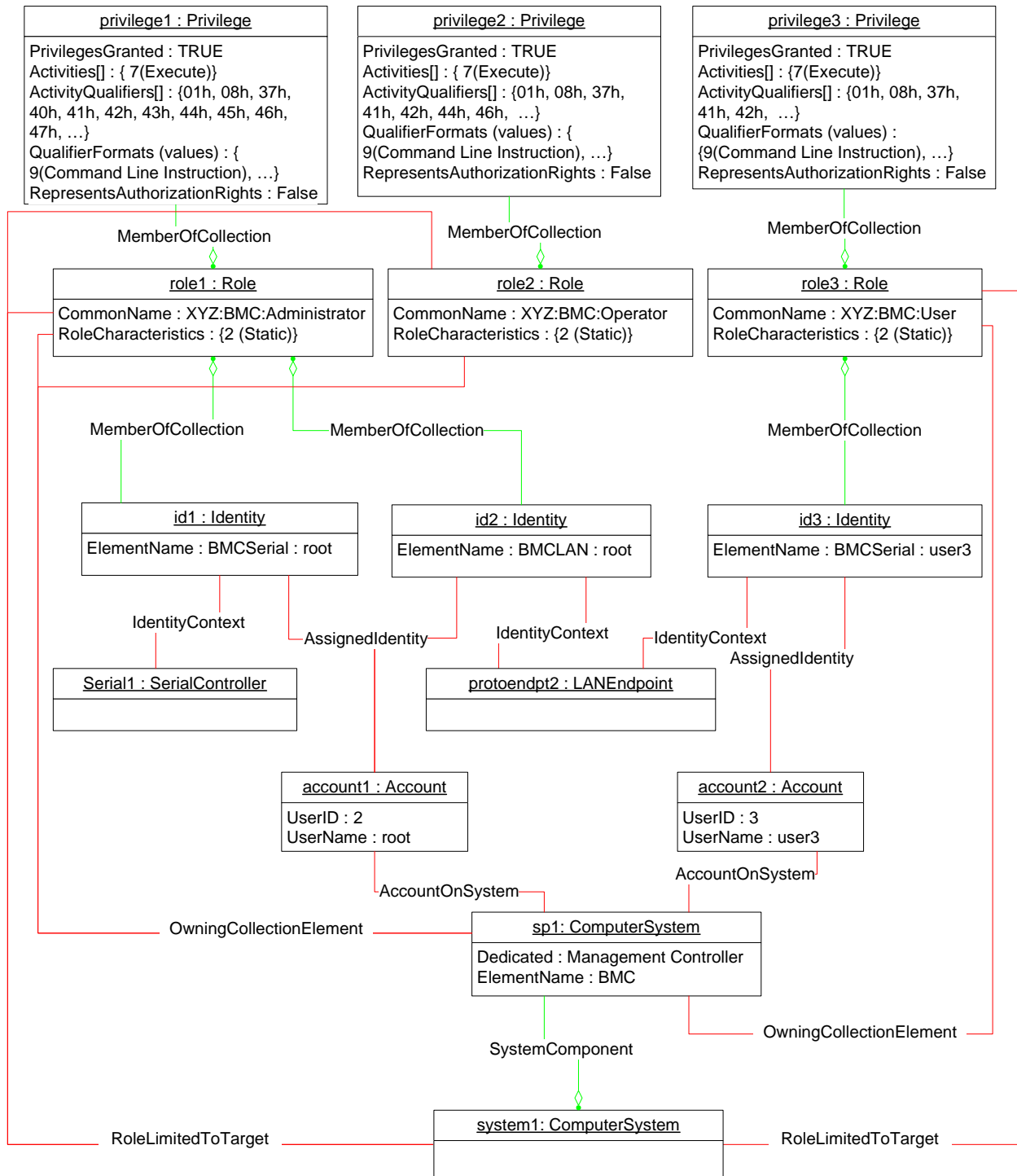
1037

**Figure 7 – Fixed Accounts with Individual Account Privilege Management**

1038 **9.5 Service Processor Roles Use Cases**

1039 This section provides object diagrams for a possible implementation of authorized roles for a service  
 1040 processor.

1041 Figure 8 represents a possible instantiation of the *Role Based Authorization Profile* for IPMI-based  
 1042 service processor roles. Three roles are represented: role1, role2, role3. These roles have the scope that  
 1043 includes system1 and the service processor, sp1. The privileges for the authorized roles are represented  
 1044 through the IPMI commands that each role allows the associated user to execute. The security principals  
 1045 id1, id2, and id3, are each associated with Serial1, protoendpt2, and protoendpt2, respectively,  
 1046 representing the communication channel that has handled the authentication. id1, id2, and id3 have  
 1047 privileges to act within system1 as denoted by the instances of CIM\_RoleLimitedToTarget that associate  
 1048 their member roles to system1. Because sp1 is a component of system1, id1, id2, and id3 have the same  
 1049 privileges within sp1.



1050

1051

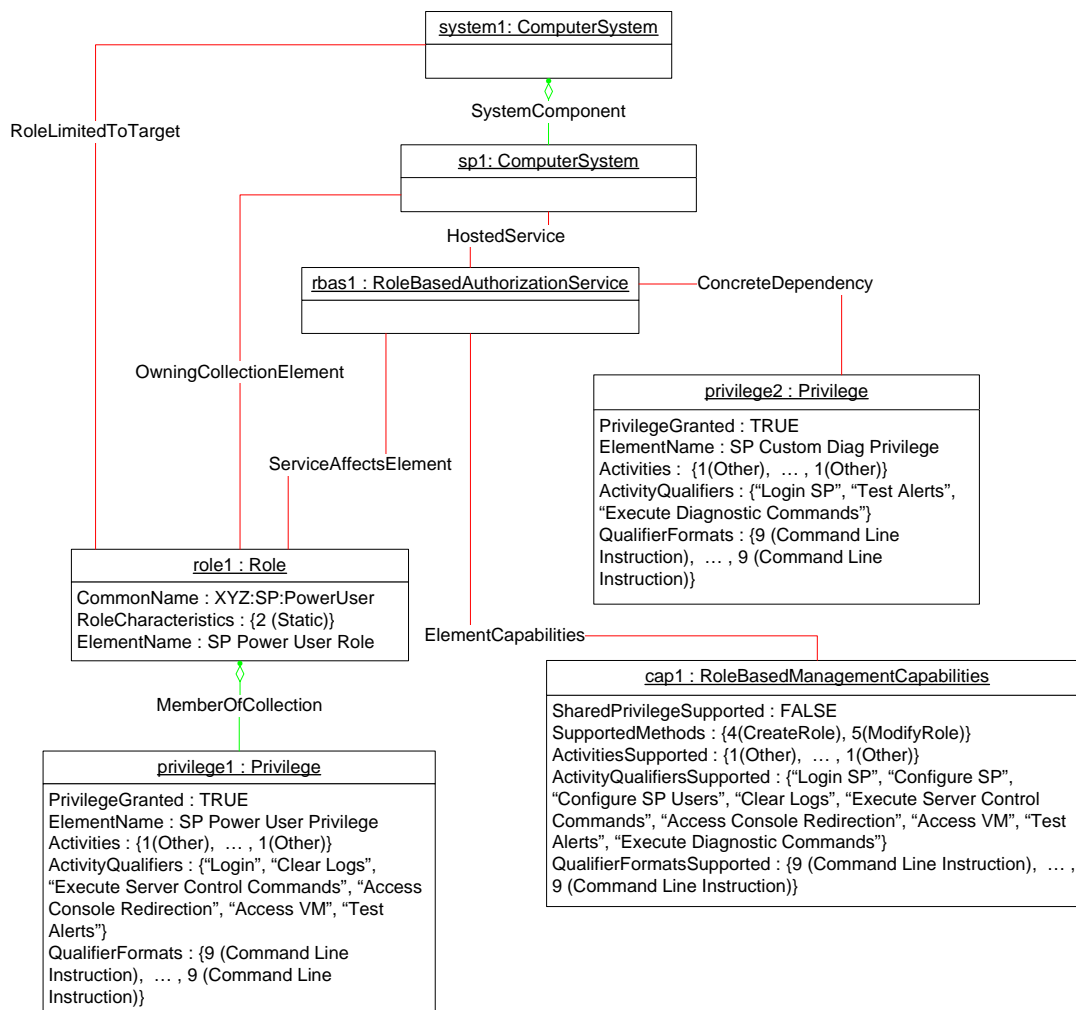
Figure 8 – IPMI Service Processor with Role Management



1052 **EXPERIMENTAL**

1053 Figure 9 represents another instantiation of the *Role Based Authorization Profile* for service processor  
 1054 roles. system1 hosts sp1, which represents the service processor. sp1 has a predefined role, role1, scope  
 1055 extends to the host computer system, system1, and the service processor itself, sp1. role1’s privileges  
 1056 are represented by privilege1. cap1 advertises the capabilities for the client to do Authorized Role  
 1057 Management. cap1’s SupportedMethods property contains two values: 4 (CreateRole) and 5  
 1058 (ModifyRole), which advertises to the client that Authorized Role Management is supported with  
 1059 CreateRole() and ModifyRole() extrinsic methods.

1060 To execute the CreateRole() method successfully, the client needs to know the type of privileges that the  
 1061 new role can support. Because the underlying device has binary representation of activities, the  
 1062 implementation has populated the ActivitiesSupported, ActivityQualifiersSupported, and  
 1063 QualifierFormatsSupported properties of cap1, and the instrumentation has instantiated a Template  
 1064 Privilege, privilege2, to give the client further guidance on the construction of the Privileges parameter of  
 1065 the CreateRole() method of rbas1.



1066

1067

**Figure 9 – IPMI Service Processor with Role Management**

1068 **EXPERIMENTAL**

## 1069 9.6 Determine the Roles Managed by a Service

1070 Given an instance of CIM\_RoleBasedAuthorizationService, a client can determine the instances of  
1071 CIM\_Role managed by the instance of CIM\_RoleBasedAuthorizationService as follows:

- 1072 1) Find the instance of CIM\_RoleBasedManagementCapabilities that is associated with the target  
1073 instance through an instance of CIM\_ElementCapabilities.
- 1074 2) If the CIM\_RoleBasedManagementCapabilities.SupportedMethods property contains the value 7  
1075 (ShowRoles), invoke the CIM\_RoleBasedAuthorizationService.ShowRoles() method, specifying Null  
1076 for the Subject and Target parameters.

1077 Upon successful completion, the Roles parameter will contain an embedded instance of CIM\_Role  
1078 for each CIM\_Role instance managed by the service.

- 1079 3) If the CIM\_RoleBasedManagementCapabilities.SupportedMethods property does not contain the  
1080 value 7 (ShowRoles), find the instances of CIM\_Role that are associated through the  
1081 CIM\_ServiceAffectsElement association.

## 1082 9.7 Determine Candidate Roles for a Security Principal

1083 Given an instance of CIM\_Identity that represents a security principal, a client can determine all of the  
1084 instances of CIM\_Role to which the CIM\_Identity instance could be assigned as follows:

- 1085 1) Find the instance of CIM\_AccountManagementService that is associated with the CIM\_Identity  
1086 instance through the CIM\_ServiceAffectsElement association.
- 1087 2) Find the instances of CIM\_RoleBasedAuthorizationService that are associated with the  
1088 CIM\_AccountManagementService through the CIM\_ServiceServiceDependency association.
- 1089 3) For each instance of CIM\_RoleBasedAuthorizationService, use the steps in section 9.6 to find the  
1090 instances of CIM\_Role that are managed by the service.

1091 The union of the instances of CIM\_Role from step 3) form the set of instances of CIM\_Role to which  
1092 the CIM\_Identity instance could be assigned.

## 1093 9.8 Determine the Roles to Which a Security Principal Is Currently Assigned

1094 Given an instance of CIM\_Identity that represents a security principal, a client can determine the  
1095 instances of CIM\_Role to which the CIM\_Identity instance is currently assigned as follows:

- 1096 1) Find the instance of CIM\_AccountManagementService that is associated with the CIM\_Identity  
1097 instance through the CIM\_ServiceAffectsElement association.
- 1098 2) Find the instances of CIM\_RoleBasedAuthorizationService that are associated with the  
1099 CIM\_AccountManagementService through the CIM\_ServiceServiceDependency association.
- 1100 3) For each instance of CIM\_RoleBasedAuthorizationService, find the instance of  
1101 CIM\_RoleBasedManagementCapabilities that is associated through the CIM\_ElementCapabilities  
1102 association.
- 1103 4) If the CIM\_RoleBasedManagementCapabilities.SupportedMethods property contains the value  
1104 7 (ShowRoles),
  - 1105 1) Invoke the CIM\_RoleBasedAuthorizationService.ShowRoles() method, specifying a reference  
1106 to the CIM\_Identity instance as the value of the Subject parameter and Null for the Target  
1107 parameter.
  - 1108 2) Upon successful completion, the Roles parameter will contain an embedded instance of  
1109 CIM\_Role for each CIM\_Role instance managed by the service, and the Privileges parameter  
1110 will contain an instance of Cumulative Privilege for each returned instance of CIM\_Role.

- 1111 5) Else, if the CIM\_RoleBasedManagementCapabilities.SupportedMethods property does not contain  
1112 the value 7 (ShowRoles), find all of the instances of CIM\_Role that are associated with the  
1113 CIM\_Identity instance through the CIM\_MemberOfCollection association.

## 1114 9.9 Determine the Roles that Scope a Managed Element

1115 Given an instance of CIM\_ManagedElement, a client can determine the instances of CIM\_Role that  
1116 scope the target instance as follows:

- 1117 1) Enumerate all the instances of CIM\_RoleBasedAuthorizationService.
- 1118 2) For each instrumented instance of CIM\_RoleBasedAuthorizationService, find the instance of  
1119 CIM\_RoleBasedManagementCapabilities that is associated through the CIM\_ElementCapabilities  
1120 association.
- 1121 3) If the CIM\_RoleBasedManagementCapabilities.SupportedMethods property contains the value  
1122 7 (ShowRoles), invoke the ShowRoles() method, specifying Null for the Subject parameter and a  
1123 reference to the CIM\_ManagedElement instance as the value of the Target parameter.
- 1124 4) Else, use the traversal algorithm described in section 7.1.1.1.

## 1125 9.10 Determine the Current Privileges of a Security Principal for a Managed 1126 Element

1127 Given an instance of CIM\_Identity that represents a security principal and an instance of  
1128 CIM\_ManagedElement, a client can determine the current privileges of the CIM\_Identity instance for  
1129 managing the instance of CIM\_ManagedElement as follows:

- 1130 1) Find the instance of CIM\_AccountManagementService that is associated with the CIM\_Identity  
1131 instance through the CIM\_ServiceAffectsElement association.
- 1132 2) Find the instances of CIM\_RoleBasedAuthorizationService that are associated with the  
1133 CIM\_AccountManagementService through the CIM\_ServiceServiceDependency association.
- 1134 3) For each instance of CIM\_RoleBasedAuthorizationService, find the instance of  
1135 CIM\_RoleBasedManagementCapabilities that is associated through the CIM\_ElementCapabilities  
1136 association.
- 1137 4) If the CIM\_RoleBasedManagementCapabilities.SupportedMethods property contains the value  
1138 1 (ShowAccess), invoke the CIM\_RoleBasedAuthorizationService.ShowAccess() method, specifying  
1139 a reference to the CIM\_Identity instance as the value of the Subject parameter and a reference to  
1140 the instance of CIM\_ManagedElement for the Target parameter.

1141 Upon successful completion, the Privileges parameter will contain an embedded instance of  
1142 CIM\_Privilege that represents the Cumulative Privilege granted to the security principal by the  
1143 instances of CIM\_Role that are managed by the instance of CIM\_RoleBasedAuthorizationService.

- 1144 5) Else, construct the Cumulative Privilege as defined in section 7.1.3.3, where the set of instances of  
1145 CIM\_Role are those instances such that the given instance of CIM\_Identity is a member of the  
1146 CIM\_Role instance as defined in section 7.3.2, and the given instance of CIM\_ManagedElement  
1147 specified by the Target parameter is in the scope of the CIM\_Role instance as defined in section  
1148 7.1.1.1.

1149

## 1150 9.11 Modify a Single Privilege of an Existing Role

1151 For a given instance of CIM\_Role that represents an existing role, a client can modify a single privilege of  
1152 the role as follows:

- 1153 1) If the RoleCharacteristics property of the selected instance of CIM\_Role does not have the value 2  
1154 (Static), then select the Associated Role Management Capability of the selected CIM\_Role instance,
  - 1155 1) If the SupportedMethods property of the Associated Privilege Management Capability of the  
1156 selected CIM\_Privilege instance has a value of 8 (ModifyPrivilege),
    - 1157 1) Execute the ModifyInstance operation on the selected instances of CIM\_Privilege,  
1158 modifying the privilege accordingly.
  - 1159 2) Else, the privileges cannot be modified.
- 1160 2) Else, the role is static and its privileges cannot be modified.

---

### 1161 EXPERIMENTAL

## 1162 9.12 Create a New Role

1163 For a given instance of CIM\_RoleBasedAuthorizationService, a client can create a new role as follows:

- 1164 1) Find the CIM\_RoleBasedManagementCapabilities instance associated to the given instance of  
1165 CIM\_RoleBasedAuthorizationService.
- 1166 2) If the SupportedMethods property of the CIM\_RoleBasedManagementCapabilities instance has a  
1167 value of 4 (CreateRole),
  - 1168 1) Construct the parameters for the CIM\_RoleBasedAuthorizationService.CreateRole() method in  
1169 the following way:
    - 1170 • RoleTemplate: Construct the desired embedded instance of CIM\_Role.
    - 1171 • OwningSystem: Construct the CIM reference to the instance of CIM\_ComputerSystem that  
1172 will be the Scoping Instance of the newly created instance of CIM\_Role.
    - 1173 • Privileges: Construct the embedded instance of CIM\_Privilege based on the  
1174 ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported  
1175 properties of the selected instance of CIM\_RoleBasedManagementCapabilities, or based  
1176 on the Template Privilege associated with the CIM\_RoleBasedAuthorizationService  
1177 instance.
    - 1178 • RoleLimitedToTargets: Construct the CIM reference to the instance of subclass of  
1179 CIM\_ManagedElement which will be the Root Instance of the new instance of CIM\_Role.
  - 1180 2) Execute the CIM\_RoleBasedAuthorizationService.CreateRole() method with the preceding  
1181 parameters.
- 1182 3) Else, the given instance of CIM\_RoleBasedAuthorizationService does not support the creation of  
1183 new role and this use case is not supported.

---

### 1184 EXPERIMENTAL

## 1185 9.13 Determine Whether Privilege Management Is Supported for a Principal

1186 A client can determine whether privilege management is supported for a security principal as follows:

- 1187 1) Starting at the instance of CIM\_Identity that represents the security principal, find the instances of  
1188 CIM\_AccountManagementService that are associated through the CIM\_ServiceAffectsElement  
1189 association.

- 1190 2) For each instance of CIM\_AccountManagementService, determine if at least one instance of  
1191 CIM\_RoleBasedAuthorizationService is associated through the CIM\_ServiceServiceDependency  
1192 association.
- 1193 3) If at least one instance of CIM\_RoleBasedAuthorizationService is associated with at least one  
1194 instance of CIM\_AccountManagementService, privilege management is supported for the security  
1195 principal.

#### 1196 **9.14 Determine Whether One-to-One Privilege Management Is Supported for an** 1197 **Account**

1198 A client can determine whether authorization for a security principal can be managed using one-to-one  
1199 correspondence as follows:

1200 Starting at the target instance of CIM\_Identity, query for an instance of CIM\_ConcreteDependency that  
1201 references the CIM\_Identity instance and an instance of CIM\_Role.

1202 If an instance exists, authorization for the CIM\_Account can be managed through one-to-one  
1203 correspondence. Note that authorization through role membership could also be supported.

#### 1204 **9.15 Assign Custom Privileges to an Identity**

1205 A client can assign custom privileges to an instance of CIM\_Account as follows:

- 1206 1) Determine whether privileges for the CIM\_Account are managed through one-to-one  
1207 correspondence or role membership as described in section 9.14.
- 1208 If privileges are not managed through one-to-one correspondence, it is necessary to create a custom  
1209 role that has the desired privileges. See section 9.12 for information about how to create a role with  
1210 the desired privileges.
- 1211 2) If privileges are managed through one-to-one correspondence, find the instance of CIM\_Identity that  
1212 is associated with the CIM\_Account instance.
- 1213 3) Find the instance of CIM\_Role that is associated with the CIM\_Identity instance through an instance  
1214 of CIM\_ConcreteDependency.
- 1215 4) If the CIM\_Identity instance is not already associated with the instance of CIM\_Role from step 3)  
1216 through an instance of CIM\_MemberOfCollection, use CreateInstance to create an instance of  
1217 CIM\_MemberOfCollection that associates the CIM\_Identity instance with the CIM\_Role instance.
- 1218 5) If the CIM\_Identity is associated with the instance of CIM\_Role other than that from step 3) through  
1219 an instance of CIM\_MemberOfCollection, use DeleteInstance to delete the instance of  
1220 CIM\_MemberOfCollection that associates the CIM\_Identity instance with the CIM\_Role instance.
- 1221 6) Perform role modification on the instance of CIM\_Role from step 3) as specified in section 9.6.

1222 **10 CIM Elements**

1223 Table 24 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be  
 1224 implemented as described in Table 24. Sections 7 (“Implementation”) and 8 (“Methods”) may impose  
 1225 additional requirements on these elements.

1226 **Table 24 – CIM Elements: Role Based Authorization Profile**

Element Name	Requirement	Description
<b>Classes</b>		
CIM_ConcreteDependency (Privilege)	Optional	See section 10.1.
CIM_ConcreteDependency (Role)	Optional	See section 10.2.
CIM_ElementCapabilities	Mandatory	See sections 7.1 and 10.3.
CIM_HostedService	Mandatory	See section 10.4.
CIM_MemberOfCollection (Privilege)	Optional	See section 10.5.
CIM_MemberOfCollection (Identity)	Optional	See section 10.6.
CIM_OwningCollectionElement	Mandatory	See section 10.7.
CIM_Privilege	Optional	See section 10.8.
CIM_RoleBasedManagementCapabilities	Mandatory	See sections 7.1 and 10.9.
CIM_RegisteredProfile	Mandatory	See section 10.10.
CIM_Role	Mandatory	See section 10.11.
CIM_RoleBasedAuthorizationService	Mandatory	See sections 7.2 and 10.12.
CIM_RoleLimitedToTarget	Mandatory	See section 10.13.
CIM_ServiceAffectsElement – CIM_Role	Mandatory	See section 10.14.
CIM_ServiceAffectsElement – CIM_Privilege	Optional	See section 10.15.
CIM_ServiceServiceDependency	Optional	See section 10.16.
<b>Indications</b>		
None defined in this profile		

1227 **10.1 CIM\_ConcreteDependency (Privilege)**

1228 CIM\_ConcreteDependency is used to associate a Template Privilege with an instance of  
 1229 CIM\_RoleBasedAuthorizationService. Table 25 contains the requirements for elements of this class.

1230 **Table 25 – Class: CIM\_ConcreteDependency (Privilege)**

Elements	Requirement	Notes
Antecedent	Mandatory	Key: This property shall reference an instance of CIM_RoleBasedAuthorizationService. Cardinality * indicating zero or more references.
Dependent	Mandatory	Key: This property shall reference a Template Privilege. Cardinality * indicating zero or more references.

1231 **10.2 CIM\_ConcreteDependency (Role)**

1232 CIM\_ConcreteDependency is used to associate an instance of CIM\_Identity with an instance of  
 1233 CIM\_Role. Table 26 contains the requirements for elements of this class.

1234 **Table 26 – Class: CIM\_ConcreteDependency (Role)**

Elements	Requirement	Notes
Antecedent	Mandatory	This property shall be a reference to CIM_Identity. Cardinality 0..1
Dependent	Mandatory	This property shall be a reference to CIM_Role. Cardinality 0..1

1235 **10.3 CIM\_ElementCapabilities**

1236 CIM\_ElementCapabilities is used to associate an instance of CIM\_RoleBasedAuthorizationService with  
 1237 an instance of CIM\_RoleBasedManagementCapabilities that describes the capabilities of the role  
 1238 management. Table 27 contains the requirements for elements of this class.

1239 **Table 27 – Class: CIM\_ElementCapabilities**

Elements	Requirement	Notes
ManagedElement	Mandatory	Key: This property shall reference the instance of CIM_RoleBasedAuthorizationService. Cardinality 1..*
Capabilities	Mandatory	Key: This property shall reference the instance of CIM_RoleBasedManagementCapabilities. Cardinality 1 indicating one and only one reference.

1240 **10.4 CIM\_HostedService**

1241 CIM\_HostedService is used to associate an instance of CIM\_RoleBasedAuthorizationService with an  
 1242 instance of CIM\_ComputerSystem that is the computer system hosting the service. Table 28 contains the  
 1243 requirements for elements of this class.

1244 **Table 28 – Class: CIM\_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	Key: This property shall reference the instance of CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	Key: This property shall reference the instance of CIM_RoleBasedAuthorizationService. Cardinality 1..*

1245 **10.5 CIM\_MemberOfCollection (Privilege)**

1246 CIM\_MemberOfCollection is used to associate an instance of CIM\_Privilege with an instance of  
 1247 CIM\_Role that represents the role that contains the privilege. Table 29 contains the requirements for  
 1248 elements of this class.

1249 **Table 29 – Class: CIM\_MemberOfCollection (Privilege)**

Elements	Requirement	Notes
Collection	Mandatory	Key: This property shall reference the instance of CIM_Role. Cardinality * indicating zero or more references.
Member	Mandatory	Key: This property shall reference the instance of CIM_Privilege. Cardinality * indicating zero or more references.

1250 **10.6 CIM\_MemberOfCollection (Identity)**

1251 Table 30 contains the requirements for instances of CIM\_MemberOfCollection if it is used to associate  
 1252 instances of CIM\_Identity with instances of CIM\_Role.

1253 **Table 30 – Class: CIM\_MemberOfCollection (Identity)**

Elements	Requirement	Notes
Collection	Mandatory	The value of this property shall be an instance of CIM_Role. Cardinality *
Member	Mandatory	This property shall be a reference to an instance of CIM_Identity. Cardinality *

1254 **10.7 CIM\_OwningCollectionElement**

1255 CIM\_OwningCollectionElement is used to associate an instance of CIM\_Role with an instance of  
 1256 CIM\_ComputerSystem that represents the computer system to which the role belongs. Table 31 contains  
 1257 the requirements for elements of this class.

1258 **Table 31 – Class: CIM\_OwningCollectionElement**

Elements	Requirement	Notes
OwningElement	Mandatory	Key: This property shall reference the instance of CIM_ComputerSystem. Cardinality 1 indicating one and only one reference.
OwnedElement	Mandatory	Key: This property shall reference the instance of CIM_Role. Cardinality 1..* indicating one or more references.



1259 **10.8 CIM\_Privilege**

1260 CIM\_Privilege is used to represent the privileges of a role. Table 32 contains the requirements for  
 1261 elements of this class.

1262 **Table 32 – Class: CIM\_Privilege**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
RepresentsAuthorizationRights	Mandatory	None
PrivilegeGranted	Mandatory	See section 7.1.3.1.
Activities	Conditional	See section 7.4.1.2.
ActivityQualifiers	Conditional	See section 7.4.1.2.
QualifierFormats	Conditional	See section 7.4.1.2.

1263 **10.9 CIM\_RoleBasedManagementCapabilities**

1264 CIM\_RoleBasedManagementCapabilities is used to indicate the capabilities for role-based privilege  
 1265 management. Table 33 contains the requirements for elements of this class.

1266 **Table 33 – Class: CIM\_RoleBasedManagementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
SharedPrivilegeSupported	Mandatory	See section 7.4.1.1.
ActivitiesSupported	Conditional	See section 7.4.1.2.
ActivityQualifiersSupported	Conditional	See section 7.4.1.2.
QualifierFormatsSupported	Optional	See section 7.4.1.2.
SupportedMethods	Mandatory	None
ElementName	Mandatory	Matches (pattern “.*”)

1267 **10.10 CIM\_RegisteredProfile**

1268 The CIM\_RegisteredProfile class is defined by the *Profile Registration Profile*. The requirements denoted  
 1269 in Table 34 are in addition to those mandated by the *Profile Registration Profile*.

1270 **Table 34 – Class: CIM\_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of “Role Based Authorization”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (“DMTF”).

1271 **10.11 CIM\_Role**

1272 CIM\_Role is used to represent an authorized role. Table 35 contains the requirements for elements of this  
 1273 class.

1274 **Table 35 – Class: CIM\_Role**

Elements	Requirement	Notes
CreationClassName	Mandatory	Key
Name	Mandatory	Key
RoleCharacteristics	Mandatory	See section 7.1.4.
CommonName	Mandatory	See section 7.1.2.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).

1275 **10.12 CIM\_RoleBasedAuthorizationService**

1276 CIM\_RoleBasedAuthorizationService is used to represent the service that handles the role management.  
 1277 Table 36 contains the requirements for elements of this class.

1278 **Table 36 – Class: CIM\_RoleBasedAuthorizationService**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “.*”).
CreateRole( )	Conditional	EXPERIMENTAL. See section 8.1.
DeleteRole( )	Conditional	EXPERIMENTAL. See section 8.2.
ModifyRole( )	Conditional	See section 8.3.
AssignRoles( )	Conditional	See section 8.4.
ShowAccess( )	Conditional	This method should be supported; see section 8.5.
ShowRoles( )	Conditional	This method should be supported; see section 8.6.

1279 **10.13 CIM\_RoleLimitedToTarget**

1280 CIM\_RoleLimitedToTarget is used to associate an instance of CIM\_Role with an instance of  
 1281 CIM\_ManagedElement that limits the scope of the role. Table 37 contains the requirements for elements  
 1282 of this class.

1283 **Table 37 – Class: CIM\_RoleLimitedToTarget**

Elements	Requirement	Notes
DefiningRole	Mandatory	Key: This property shall reference the instance of CIM_Role. Cardinality * indicating zero or more references.
TargetElement	Mandatory	Key: This property shall reference the instance of CIM_ManagedElement. Cardinality 1..*

1284 **10.14 CIM\_ServiceAffectsElement – CIM\_Role**

1285 CIM\_ServiceAffectsElement is used to associate an instance of CIM\_RoleBasedAuthorizationService  
 1286 with an instance of CIM\_Role that represents the role that could be modified by using the service. Table  
 1287 38 contains the requirements for elements of this class.

1288 **Table 38 – Class: CIM\_ServiceAffectsElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	Key: This property shall reference the instance of CIM_Role. Cardinality 1..*
AffectingElement	Mandatory	Key: This property shall reference the instance of CIM_RoleBasedAuthorizationService. Cardinality 1
ElementEffects	Mandatory	Matches 5 (Manages)

1289 **10.15 CIM\_ServiceAffectsElement – CIM\_Privilege**

1290 If the instance of CIM\_Privilege is associated with instances of CIM\_Role which are in turn associated  
 1291 with different instances of CIM\_RoleBasedAuthorizationService, CIM\_ServiceAffectsElement associating  
 1292 CIM\_Privilege with a CIM\_RoleBasedAuthorizationService instance shall be implemented.

1293 CIM\_ServiceAffectsElement is used to associate an instance of CIM\_RoleBasedAuthorizationService  
 1294 with an instance of CIM\_Privilege that represents a privilege. Table 39 contains the requirements for  
 1295 elements of this class.

1296 **Table 39 – Class: CIM\_ServiceAffectsElement**

Elements	Requirement	Notes
AffectedElement	Mandatory	Key: This property shall reference the instance of CIM_Privilege. Cardinality 1..*
AffectingElement	Mandatory	Key: This property shall reference the instance of CIM_RoleBasedAuthorizationService. Cardinality 1
ElementEffects	Mandatory	Matches 5 (Manages)

1297 **10.16 CIM\_ServiceServiceDependency**

1298 CIM\_ServiceServiceDependency is used to associate an instance of  
 1299 CIM\_RoleBasedAuthorizationService with an instance of CIM\_AccountManagementService representing  
 1300 that the identities of the CIM\_AccountManagementService instance could be members of roles of the  
 1301 associated CIM\_RoleBasedAuthorizationService instance. Table 40 contains the requirements for  
 1302 elements of this class.

1303

**Table 40 – Class: CIM\_ServiceServiceDependency**

Elements	Requirement	Notes
Antecedent	Mandatory	Key: This property shall be a reference to an instance of CIM_AccountManagementService. Cardinality *
Dependent	Mandatory	Key: This property shall be a reference to the Central Instance of the profile. Cardinality *
TypeOfDependency	Mandatory	Matches 5 (Cooperate)

**ANNEX A  
(informative)**

**Change Log**

1304  
1305  
1306  
1307  
1308

Version	Date	Author	Description
1.0.0a	2006/10/23	Khachatur Papanyan	Preliminary Standard version.
1.0.0	2008/07/03	Khachatur Papanyan	Final version.

**ANNEX B  
(informative)**

1309  
1310  
1311  
1312  
1313

**Acknowledgements**

1314 The authors wish to acknowledge the following people.

1315 Editors:

- 1316 • Khachatur Papanyan – Dell
- 1317 • Aaron Merkin – IBM

1318 Contributors:

- 1319 • Murali Rajagopal – Broadcom
- 1320 • Hemal Shah – Broadcom
- 1321 • Jon Hass – Dell
- 1322 • Khachatur Papanyan – Dell
- 1323 • George Ericson – EMC
- 1324 • Christina Shaw – HP
- 1325 • Aaron Merkin – IBM
- 1326 • David Hines – Intel