



1
2
3
4

Document Number: DSP1039

Date: 2006-10-23

Version: 1.0.0a

5 **Role Based Authorization Profile**

6 **Document Type: Specification**
7 **Document Status: Preliminary Standard**
8 **Document Language: E**
9

Role Based Authorization Profile

10 Copyright notice

11 Copyright © 2006 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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

31 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 12

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 (Optional) 18

50 7.3 Authorized Role Membership of Security Principal (Optional)..... 19

51 8 Methods..... 20

52 8.1 CIM_RoleBasedAuthorizationService.CreateRole() 20

53 8.2 CIM_RoleBasedAuthorizationService.DeleteRole() 22

54 8.3 CIM_RoleBasedAuthorizationService.ModifyRole() 23

55 8.4 CIM_RoleBasedAuthorizationService.AssignRoles() 24

56 8.5 CIM_RoleBasedAuthorizationService.ShowAccess() 25

57 8.6 CIM_RoleBasedAuthorizationService.ShowRoles() 26

58 8.7 Profile Conventions for Operations 28

59 8.8 CIM_ConcreteDependency 29

60 8.9 CIM_ElementCapabilities 29

61 8.10 CIM_HostedService 29

62 8.11 CIM_MemberOfCollection 30

63 8.12 CIM_OwningCollectionElement 30

64 8.13 CIM_Privilege..... 30

65 8.14 CIM_RoleBasedManagementCapabilities 31

66 8.15 CIM_Role 31

67 8.16 CIM_RoleBasedAuthorizationService..... 31

68 8.17 CIM_RoleLimitedToTarget..... 31

69 8.18 CIM_ServiceAffectsElement 31

70 8.19 CIM_ServiceServiceDependency 32

71 9 Use Cases..... 32

72 9.1 Profile Registration..... 32

73 9.2 Minimal Instantiation of the Profile..... 33

74 9.3 Evaluating Scope and Privileges 33

75 9.4 Scope of the Role and Privileges for a Managed Element 37

76 9.5 Service Processor Roles Use Cases..... 39

Role Based Authorization Profile

77	9.6	Determine the Roles Managed by a Service	42
78	9.7	Determine Candidate Roles for a Security Principal	42
79	9.8	Determine the Roles to Which a Security Principal Is Currently Assigned.....	42
80	9.9	Determine the Roles that Scope a Managed Element	43
81	9.10	Determine the Current Privileges of a Security Principal for a Managed Element.....	43
82	9.11	Modify the Privileges of an Existing Role.....	43
83	9.12	Create a New Role.....	43
84	9.13	Determine Whether Privilege Management Is Supported for a Principal	44
85	9.14	Determine Whether One-to-One Privilege Management Is Supported for an Account.....	44
86	9.15	Assign Custom Privileges to an Identity	44
87	10	CIM Elements	45
88	10.1	CIM_ConcreteDependency (Privilege)	46
89	10.2	CIM_ConcreteDependency (Role).....	46
90	10.3	CIM_ElementCapabilities	46
91	10.4	CIM_HostedService	47
92	10.5	CIM_MemberOfCollection (Privilege)	47
93	10.6	CIM_MemberOfCollection (Identity)	47
94	10.7	CIM_OwningCollectionElement	48
95	10.8	CIM_Privilege.....	48
96	10.9	CIM_RoleBasedManagementCapabilities	48
97	10.10	CIM_RegisteredProfile.....	49
98	10.11	CIM_Role	49
99	10.12	CIM_RoleBasedAuthorizationService.....	49
100	10.13	CIM_RoleLimitedToTarget.....	50
101	10.14	CIM_ServiceAffectsElement	50
102	10.15	CIM_ServiceServiceDependency	50
103	ANNEX A (informative)	Change Log.....	51
104	ANNEX B (informative)	Acknowledgements	52
105			
106	Figures		
107	Figure 1 – Role Based Authorization Profile: Class Diagram		13
108	Figure 2 – Profile Registration.....		32
109	Figure 3 – Minimal Instantiation		33
110	Figure 4 – Cumulative Role Privilege Example.....		34
111	Figure 5 – Roles and Privileges for Principals		37
112	Figure 6 – Fixed Accounts with Role Membership Privilege Management		38
113	Figure 7 – Fixed Accounts with Individual Account Privilege Management		39
114	Figure 8 – IPMI Service Processor with Role Management		40
115	Figure 9 – IPMI Service Processor with Role Management		41
116			
117	Tables		
118	Table 1 – Referenced Profiles		12
119	Table 2 – Containment Relationships		15
120	Table 3 – CIM_RoleBasedAuthorizationService.CreateRole() Method: Return Code Values		21
121	Table 4 – CIM_RoleBasedAuthorizationService.CreateRole() Method: Parameters		21
122	Table 5 – CIM_RoleBasedAuthorizationService.DeleteRole() Method: Return Code Values.....		23
123	Table 6 – CIM_RoleBasedAuthorizationService.DeleteRole() Method: Parameters		23

Role Based Authorization Profile

124 Table 7 – CIM_RoleBasedAuthorizationService.ModifyRole() Method: Return Code Values 24

125 Table 8 – CIM_RoleBasedAuthorizationService.ModifyRole() Method: Parameters 24

126 Table 9 – CIM_RoleBasedAuthorizationService.AssignRoles() Method: Return Code Values 25

127 Table 10 – CIM_RoleBasedAuthorizationService.AssignRoles() Method: Parameters 25

128 Table 11 – CIM_RoleBasedAuthorizationService.ShowAccess() Method: Return Code Values 26

129 Table 12 – CIM_RoleBasedAuthorizationService.ShowAccess() Method: Parameters 26

130 Table 13 – CIM_RoleBasedAuthorizationService.ShowRoles() Method: Return Code Values 27

131 Table 14 – CIM_RoleBasedAuthorizationService.ShowRoles() Method: Parameters 28

132 Table 15 – Operations: CIM_ConcreteDependency 29

133 Table 16 – Operations: CIM_ElementCapabilities 29

134 Table 17 – Operations: CIM_HostedService 29

135 Table 18 – Operations: CIM_MemberOfCollection 30

136 Table 19 – Operations: CIM_OwningCollectionElement 30

137 Table 20 – Operations: CIM_Privilege 30

138 Table 21 – Operations: CIM_RoleLimitedToTarget 31

139 Table 22 – Operations: CIM_ServiceAffectsElement 31

140 Table 23 – Operations: CIM_ServiceServiceDependency 32

141 Table 24 – CIM Elements: Role Based Authorization Profile 45

142 Table 25 – Class: CIM_ConcreteDependency (Privilege) 46

143 Table 26 – Class: CIM_ConcreteDependency (Role) 46

144 Table 27 – Class: CIM_ElementCapabilities 46

145 Table 28 – Class: CIM_HostedService 47

146 Table 29 – Class: CIM_MemberOfCollection (Privilege) 47

147 Table 30 – Class: CIM_MemberOfCollection (Identity) 47

148 Table 31 – Class: CIM_OwningCollectionElement 48

149 Table 32 – Class: CIM_Privilege 48

150 Table 33 – Class: CIM_RoleBasedManagementCapabilities 48

151 Table 34 – Class: CIM_RegisteredProfile 49

152 Table 35 – Class: CIM_Role 49

153 Table 36 – Class: CIM_RoleBasedAuthorizationService 49

154 Table 37 – Class: CIM_RoleLimitedToTarget 50

155 Table 38 – Class: CIM_ServiceAffectsElement 50

156 Table 39 – Class: CIM_ServiceServiceDependency 50

157

Foreword

159 The *Role Based Authorization Profile* (DSP1039) was prepared by the Server Management Working
160 Group and WBEM Infrastructure and Protocols Working Group of the WBEM Infrastructures and
161 Protocols.

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

164

Introduction

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

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

174

Role Based Authorization Profile

175 1 Scope

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

182 2 Normative References

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

186 2.1 Approved References

- 187 DMTF [DSP0200](#), *CIM Operations over HTTP 1.2.0*
188 DMTF [DSP0004](#), *CIM Infrastructure Specification 2.3.0*
189 DMTF [DSP1000](#), *Management Profile Specification Template*
190 DMTF [DSP1001](#), *Management Profile Specification Usage Guide*

191 2.2 References under Development

- 192 DMTF DSP1034, *Simple Identity Management Profile 1.0*
193 DMTF [DSP1033](#), *Profile Registration Profile 1.0*
194 DMTF [DSP0215](#), *Server Management Managed Element Addressing Specification, 1.0.0*

195 2.3 Other References

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

199 3 Terms and Definitions

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

202 3.1

203 can

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

205 **3.2**
206 **cannot**
207 used for statements of possibility and capability, whether material, physical, or causal

208 **3.3**
209 **conditional**
210 indicates requirements to be followed strictly to conform to the document when the specified conditions
211 are met

212 **3.4**
213 **mandatory**
214 indicates requirements to be followed strictly to conform to the document and from which no deviation is
215 permitted

216 **3.5**
217 **may**
218 indicates a course of action permissible within the limits of the document

219 **3.6**
220 **need not**
221 indicates a course of action permissible within the limits of the document

222 **3.7**
223 **optional**
224 indicates a course of action permissible within the limits of the document

225 **3.8**
226 **referencing profile**
227 indicates a profile that owns the definition of this class and can include a reference to this profile in its
228 "Referenced Profiles" table

229 **3.9**
230 **shall**
231 indicates requirements to be followed strictly to conform to the document and from which no deviation is
232 permitted

233 **3.10**
234 **shall not**
235 indicates requirements to be followed strictly to conform to the document and from which no deviation is
236 permitted

237 **3.11**
238 **should**
239 indicates that among several possibilities, one is recommended as particularly suitable, without
240 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

241 **3.12**
242 **should not**
243 indicates that a certain possibility or course of action is deprecated but not prohibited

- 244 **3.13**
 245 **unspecified**
 246 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 247 **3.14**
 248 **Associated Privilege Management Capability**
 249 an instance of CIM_RoleBasedManagementCapabilities, which is associated with the instance of
 250 CIM_RoleBasedAuthorizationService through the CIM_ElementCapabilities association, which in turn is
 251 associated with the mentioned instance of CIM_Privilege through the CIM_ServiceAffectsElement
 252 association
- 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 None

287 **5 Synopsis**

288 **Profile Name:** *Role Based Authorization*

289 **Version:** 1.0.0

290 **Organization:** DMTF

291 **CIM schema version:** 2.14

292 **Central Class:** CIM_RoleBasedAuthorizationService

293 **Scoping Class:** CIM_ComputerSystem

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

296 The Central Class of the *Role Based Authorization Profile* shall be CIM_RoleBasedAuthorizationService.
297 The Central Instance shall be an instance of CIM_RoleBasedAuthorizationService. The Scoping Class
298 shall be CIM_ComputerSystem. The Scoping Instance shall be the instance of CIM_ComputerSystem
299 that is associated with the Central Instance through the CIM_HostedService association.

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

301 **Table 1 – Referenced Profiles**

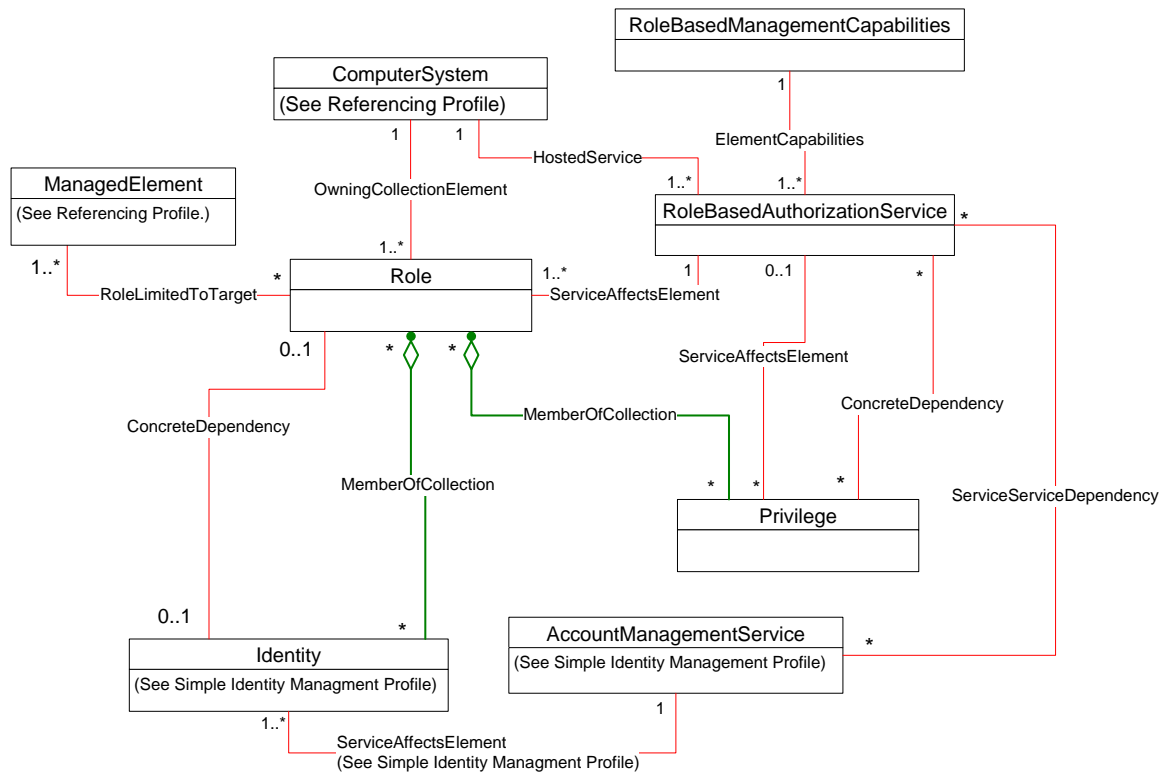
Profile Name	Organization	Version	Relationship	Behavior
<i>Simple Identity Management Profile</i>	DMTF	1.0	Optional	See section 7.3.
<i>Profile Registration Profile</i>	DMTF	1.0	Mandatory	

302 **6 Description**

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

307 Figure 1 represents the class schema for the profile. For simplicity, the prefix *CIM_* has been removed
308 from the names of the classes.

Role Based Authorization Profile



309

310

Figure 1 – Role Based Authorization Profile: Class Diagram

311 6.1 Role Authorization Service: CIM_RoleBasedAuthorizationService

312 The ability to manage and configure roles for a managed system is represented by the
 313 CIM_RoleBasedAuthorizationService instance. The CIM_RoleBasedAuthorizationService class is the
 314 Central Class of the profile and, through extrinsic methods, serves as the interface for a client to request
 315 deletion and modification of existing roles, creation of new roles, and assignment of roles to security
 316 principals.

317 6.2 Authorized Roles and Privileges: CIM_Role and CIM_Privilege

318 The authorized roles on a managed system are represented through instances of CIM_Role. Rights
 319 granted to a security principal through membership in a role are represented by instances of
 320 CIM_Privilege that are associated with the instance of CIM_Role through the CIM_MemberOfCollection
 321 association.

322 6.2.1 Role Privileges

323 When the security principal is a member of an authorized role, the principal is granted the cumulative
 324 privileges of the role. Every authorized role on the managed system can have a set of explicitly granted or
 325 denied privileges. The PrivilegeGranted property of the CIM_Privilege instance represents whether the
 326 instance of CIM_Privilege comprises activities that are granted or denied for the role. The Activities,
 327 ActivityQualifiers, and QualifierFormats properties of the CIM_Privilege instance describe the activities
 328 represented by the privilege.

329 **6.2.2 Role Scope**

330 The scope of the authorized role is the set of managed elements represented by the instances of the
331 CIM_ManagedElement subclass, which could be subjected to the activities that make up the privileges of
332 the authorized role. The scope of the roles authorization is represented by associating the CIM_Role
333 instance to instances of CIM_ManagedElement through the CIM_RoleLimitedToTarget association. When
334 the associated CIM_ManagedElement instance contains or aggregates additional CIM_ManagedElement
335 instances, the privileges granted by the role can propagate to the contained or aggregated instances of
336 CIM_ManagedElement. This profile does not provide a mechanism for managing whether the privileges
337 granted by an instance of CIM_Role for managing an instance of CIM_ManagedElement are propagated
338 to aggregated or contained instances of CIM_ManagedElement. Therefore, privileges granted for
339 managing or accessing an instance of CIM_ManagedElement always propagate to the aggregated and
340 contained instances of CIM_ManagedElement.

341 The detailed requirements for representing the scope of the authorized role are described in section
342 7.1.1.

343 **6.2.3 Cumulative Privileges**

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

348 **6.3 Security Principal: CIM_Identity**

349 The CIM_Identity class represents the security principal for the accounts (CIM_Account), users
350 (CIM_UserContact), and groups (CIM_Group) as described in *the Simple Identity Management Profile*.
351 The security principal exists on the managed system and is used to provide the security context under
352 which the authenticated user and group can act within the managed system. As such, the instantiation of
353 a CIM_Identity instance that represents the security principal does not depend on the underlying
354 authentication of the associated users and groups.

355 CIM_Identity instances that represent security principals for the accounts, users, and groups can have a
356 CIM_MemberOfCollection association to the appropriate CIM_Role instances. The representation of roles
357 is described in detail in section 6.2.

358 **6.4 Privilege Management**

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

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

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

376 The CIM_ServiceServiceDependency association is used to associate instances of
 377 CIM_AccountManagementService with instances of CIM_RoleBasedAuthorizationService. This
 378 association indicates that security principals managed by the instance of
 379 CIM_AccountManagementService can be assigned to roles managed by the instance of
 380 CIM_RoleBasedAuthorizationService.

381 7 Implementation

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

384 7.1 Modeling the Authorized Role

385 The implementation shall instantiate at least one instance of CIM_Role that represents an authorized role
 386 and at least one instance of CIM_RoleBasedAuthorizationService.

387 The instance of CIM_RoleBasedAuthorizationService shall be associated with instances of CIM_Role
 388 through CIM_ServiceAffectsElement associations.

389 The instance of CIM_RoleBasedAuthorizationService shall be associated to only one instance of
 390 CIM_ComputerSystem through the CIM_HostedService association. This instance of
 391 CIM_ComputerSystem shall be the Scoping Instance.

392 The CIM_Role instance shall be associated to only one instance of CIM_ComputerSystem, through the
 393 CIM_OwningCollectionElement association.

394 7.1.1 Scope of the Authorized Role

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

397 Each instance of CIM_Role shall be referenced by at least one instance of CIM_RoleLimitedToTarget.
 398 The CIM_RoleLimitedToTarget association explicitly places the referenced instance of
 399 CIM_ManagedElement into the scope of the CIM_Role instance. Additional instances of
 400 CIM_ManagedElement may be implicitly within the scope of the CIM_Role instance.

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

403 **Table 2 – Containment Relationships**

Container Class (REF role)	Association Class	Contained Class (REF role)
CIM_ManagedElement (GroupComponent)	CIM_Component	CIM_ManagedElement (PartComponent)
CIM_ManagedElement (Antecedent)	CIM_Dependent	CIM_ManagedElement (Dependent)
CIM_Collection (Collection)	CIM_MemberOfCollection	CIM_ManagedElement (Member)
CIM_ManagedElement (OwningElement)	CIM_OwningCollectionElement	CIM_Collection (OwnedElement)

Container Class (REF role)	Association Class	Contained Class (REF role)
CIM_RecordLog (Log)	CIM_LogManagesRecord	CIM_LogRecord (Record)
CIM_System (System)	CIM_InstalledSoftwareIdentity	CIM_SoftwareIdentity (InstalledSoftware)

404 7.1.1.1 Managed Element within Role's Scope

405 This section defines the algorithm used to determine whether an instance of CIM_ManagedElement is
406 within the scope of an instance of CIM_Role.

407 An instance of CIM_ManagedElement shall be in the scope of an instance of CIM_Role when

- 408 1) The instance of CIM_ManagedElement is associated with the instance of CIM_Role through the
409 CIM_RoleLimitedToTarget association.
- 410 2) The instance of CIM_ManagedElement is referenced by an instance of an association class
411 specified in the "Association Class" column of Table 2 where a reference to the instance of
412 CIM_ManagedElement is the value of the property specified in the "Contained Class" column of
413 Table 2 and the instance of CIM_ManagedElement referenced by the property specified in the
414 "Container Class" column of Table 2 is in the scope of the instance of CIM_Role, where the scope is
415 determined by recursively applying this algorithm.

416 7.1.2 CIM_Role.CommonName

417 The CIM_Role.CommonName property shall be formatted using the following algorithm:

418 < OrgID > : < LocalID >, where < OrgID > and < LocalID > are separated by a colon (:), and where
419 < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the
420 business entity that is creating or defining the CommonName or that is a registered ID assigned to the
421 business entity by a recognized global authority. (This requirement is similar to the < Schema Name > _
422 < Class Name > structure of Schema class names.) In addition, to ensure uniqueness, < OrgID > shall
423 not contain a colon (:). When using this algorithm, the first colon to appear in this property shall appear
424 between < OrgID > and < LocalID >. < LocalID > is chosen by the business entity and should not be
425 reused to identify different underlying (real-world) elements.

426 7.1.3 Privileges of Authorized Role

427 The privileges of an authorized role may be represented by instances of CIM_Privilege. When the
428 CIM_Role.RoleCharacteristics property contains the value 3 (Opaque), no instances of CIM_Privilege
429 shall be associated with the instance of CIM_Role through the CIM_MemberOfCollection association.

430 When the CIM_Role.RoleCharacteristics property does not contain the value 3 (Opaque), zero or more
431 instances of CIM_Privilege shall be associated with the instance of CIM_Role through the
432 CIM_MemberOfCollection association.

433 The three types of CIM_Privilege instances are Denied Privileges, Granted Privileges, and Template
434 Privileges (see sections 3.14, 3.16, and 3.20).

435 7.1.3.1 Granted Privileges and Denied Privileges

436 Granted Privileges and Denied Privileges are associated with instances of CIM_Role through instances of
437 CIM_MemberOfCollection. When at least one instance of CIM_Privilege is associated with an instance of
438 CIM_Role, at least one Granted Privilege shall be associated with the instance of CIM_Role. Any
439 activities that are not represented by Granted Privileges associated with an instance of CIM_Role are
440 assumed as denied activities for the role.

441 When the instance of CIM_Role is associated with Denied Privileges and Granted Privileges, the Denied
442 Privileges shall take precedence over the Granted Privileges.

443 7.1.3.2 Cumulative Privileges for a Role

444 More than one Granted Privilege and more than one Denied Privilege can be associated with an instance
445 of CIM_Role. This section defines an algorithm to accumulate all the rights for a given role into one
446 conceptual instance of CIM_Privilege, Cumulative Role Privilege (see section 3.16). Upon completion of
447 this algorithm, the Cumulative Role Privilege will reflect the rights explicitly granted by the instance of
448 CIM_Role.

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

- 450 1) Select all the Granted Privileges (instances of CIM_Privilege with the PrivilegeGranted property set
451 to TRUE) that are associated with the given CIM_Role instance through CIM_MemberOfCollection
452 associations.
- 453 2) For each instance of Granted Privileges, select the CIM_Privilege.Activities,
454 CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats array properties.
- 455 3) For each element in the CIM_Privilege.Activities property array, select the value of the corresponding
456 index of CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and
457 CIM_Privilege.QualifierFormats property arrays,
 - 458 – Determine if the Cumulative Role Privilege's CIM_Privilege.Activities,
459 CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays contain the
460 combination of selected element values from step 3.
 - 461 – If not, add the combination of selected values to the appropriate array properties of Cumulative
462 Role Privilege.
- 463 4) Select all the Denied Privileges (instances of CIM_Privilege with the PrivilegeGranted property set to
464 FALSE) that are associated with the given CIM_Role instance through CIM_MemberOfCollection
465 associations.
- 466 5) For each instance of Denied Privileges, select the CIM_Privilege.Activities,
467 CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats array properties.
- 468 6) For each element in the CIM_Privilege.Activities property array, select the value of the corresponding
469 index of CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and
470 CIM_Privilege.QualifierFormats property arrays,
 - 471 – Determine if the Cumulative Role Privilege's CIM_Privilege.Activities,
472 CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays contain the
473 combination of selected element values.
 - 474 – If it does, remove the combination of selected values from the appropriate array properties of
475 Cumulative Role Privilege.

476 If the CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, or CIM_Privilege.QualifierFormats
477 property is Null for all instances of CIM_Privilege where the CIM_Privilege.PrivilegeGranted property has
478 the value TRUE, the property shall be Null for the Cumulative Role Privilege.

479 7.1.3.3 Cumulative Privileges for Multiple Roles

480 The Cumulative Privilege granted by the instances of CIM_Role in an arbitrary set of instances of
481 CIM_Role shall be defined as follows:

- 482 1) For each instance of CIM_Role in the set, follow the algorithm in section 7.1.3.2 to construct the
483 Cumulative Role Privileges for the instance.

- 484 2) For each instance of Cumulative Role Privileges,
485 – For each element in the CIM_Privilege.Activities property array, select the value of the
486 corresponding index of CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and
487 CIM_Privilege.QualifierFormats property arrays,
488 1) Determine if the Cumulative Privilege's CIM_Privilege.Activities,
489 CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays
490 contain the combination of selected element values from step 1.
491 2) If not, add the combination of selected values to the appropriate array properties of
492 Cumulative Role Privilege.

493 7.1.3.4 Template Privileges

494 Template Privileges are used to provide the client with guidance for the Privileges parameter of the
495 CIM_RoleBasedAuthorizationService.CreateRole() and
496 CIM_RoleBasedAuthorizationService.ModifyRole() methods. An element in the array of the Privileges
497 parameter of these methods may be created from Template Privileges by replicating all the properties of
498 a Template Privilege with the exception of keys.

499 The Template Privileges shall be associated with instances of CIM_RoleBasedAuthorizationService
500 through instances of CIM_ConcreteDependency.

501 7.1.4 Static Authorized Role

502 An authorized role that cannot be modified or deleted by the instrumentation is referred to as a static
503 authorized role. The CIM_Role.RoleCharacteristics property shall contain the value 2 (Static Role) for an
504 instance of CIM_Role that represents a static authorized role. The CIM_Role instance that represents the
505 static authorized role shall not support Authorized Role Management as described in section 7.2.

506 7.2 Authorized Role Management (Optional)

507 Authorized Role Management provides functionality for creating, deleting, and modifying instances of
508 CIM_Role, associated instances of CIM_Privilege, and necessary associations.

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

- 510 • Creation of a CIM_Role instance and associated CIM_Privilege instances by using the
511 CIM_RoleBasedAuthorizationService.CreateRole() method. See section 8.1 for requirement details.
- 512 • Deletion of a CIM_Role instance and associated CIM_Privilege instances by using the
513 CIM_RoleBasedAuthorizationService.DeleteRole() method. See section 8.1.1 for requirement
514 details.
- 515 • Modification of a CIM_Role instance and associated CIM_Privilege instances by using the
516 CIM_RoleBasedAuthorizationService.ModifyRole() method. See section 8.2.1 for requirement
517 details.
- 518 • Modification of a CIM_Privilege instance by using the ModifyInstance operation. See section 8.13 for
519 requirement details.

520 7.2.1 Authorized Role Management Support

521 Authorized Role Management shall be supported for an instance of CIM_Role when the
522 SupportedMethods property array of the Associated Role Management Capability of the CIM_Role
523 instance contains at least one value and the CIM_Role.RoleCharacteristics property does not contain the
524 value 2 (Static).

525 **7.2.2 Authorized Role Management Capabilities:** 526 **CIM_RoleBasedManagementCapabilities**

527 Exactly one instance of CIM_RoleBasedManagementCapabilities shall be associated with the
528 CIM_RoleBasedAuthorizationService instance through the CIM_ElementCapabilities association.

529 **7.2.2.1 CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported**

530 When the CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported property is set to FALSE,
531 only one instance of CIM_Role shall be associated with each instance of CIM_Privilege, where the
532 CIM_Role instance is associated with the CIM_RoleBasedAuthorizationService instance through the
533 CIM_ServiceAffectsElement association, and the CIM_RoleBasedAuthorizationService instance is
534 associated with the CIM_RoleBasedManagementCapabilities instance through the
535 CIM_ElementCapabilities association.

536 When the CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported property is set to TRUE,
537 one or more instances of CIM_Role may be associated with each instance of CIM_Privilege, where the
538 CIM_Role instances are associated with the CIM_RoleBasedAuthorizationService instance through the
539 CIM_ServiceAffectsElement association, and the CIM_RoleBasedAuthorizationService instance is
540 associated with the CIM_RoleBasedManagementCapabilities instance through the
541 CIM_ElementCapabilities association.

542 **7.2.2.2 Supported Activities (Optional)**

543 The ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties of the
544 CIM_RoleBasedManagementCapabilities class represent the full list of supported activities of the
545 privileges.

546 When the ModifyInstance operation is supported on an instance of CIM_Privilege, the
547 ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties on the
548 Associated Privilege Management Capability of the instance of CIM_Privilege shall be supported.

549 When the implementation supports the ActivitiesSupported, ActivityQualifiersSupported, and
550 QualifierFormatsSupported properties on the Associated Privilege Management Capability of the instance
551 of CIM_Privilege, the following rules apply:

- 552 • The CIM_Privilege.Activities property array shall contain a subset of elements of the
553 ActivitiesSupported property array elements.
- 554 • The CIM_Privilege.ActivityQualifiers property array shall contain a subset of elements of the
555 ActivityQualifiersSupported property array elements.
- 556 • The CIM_Privilege.QualifierFormats property array shall contain a subset of elements of the
557 QualifierFormatsSupported property array elements.

558 **7.3 Authorized Role Membership of Security Principal (Optional)**

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

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

562 **7.3.1 Roles Available to Principal**

563 For each instance of CIM_Role with which an instance of CIM_Identity may be associated through the
564 CIM_MemberOfCollection association, an instance of CIM_ServiceServiceDependency shall associate at
565 least one CIM_AccountManagementService instance that is associated through the
566 CIM_ServiceAffectsElement association with the CIM_Identity instance to the instance of

567 CIM_RoleBasedAuthorizationService that is associated through the CIM_ServiceAffectsElement
568 association to the instance of CIM_Role.

569 **7.3.2 Managing Privileges through Role Assignment**

570 Privileges for a principal may be managed by assigning the principal to zero or more roles. An instance of
571 CIM_Identity shall be a member of an instance of CIM_Role only if an instance of
572 CIM_MemberOfCollection associates the instance of CIM_Identity that represents the principal with the
573 instance of CIM_Role that represents a role assigned to the principal.

574 When the CIM_Identity instance is not associated with any instances of CIM_Role through the
575 CIM_MemberOfCollection association, the principal shall not have any privileges.

576 **7.3.3 Managing Privileges One to One for a Principal**

577 The privileges for an authenticated entity may be modeled through a one-to-one correspondence of
578 instances of CIM_Role with an instance of CIM_Identity. When privileges are managed through one-to-
579 one correspondence, the requirements specified in this section shall be met.

580 Exactly one instance of CIM_ConcreteDependency shall be implemented as defined in section 10.2 that
581 associates the CIM_Identity instance with a CIM_Role instance. At most one instance of CIM_Identity
582 shall be associated with the CIM_Role instance through the CIM_MemberOfCollection association. When
583 an instance of CIM_Identity is associated with the CIM_Role instance through the
584 CIM_MemberOfCollection association, the CIM_Identity instance shall be the same instance with which
585 the CIM_Role instance is associated through the CIM_ConcreteDependency instance. The instance
586 relationship through CIM_ConcreteDependency is used to indicate that the CIM_Role instance can be
587 used for at most the single CIM_Identity instance with which it is associated.

588 **8 Methods**

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

591 **8.1 CIM_RoleBasedAuthorizationService.CreateRole()**

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

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

- 594 • An instance of CIM_Role shall exist that is the exact replica of the embedded instance of CIM_Role
595 of the RoleTemplate parameter except for the key properties.
- 596 • An instance of the CIM_OwningCollectionElement association shall associate the new CIM_Role
597 instance and the scoping CIM_ComputerSystem instance referenced by the OwningSystem
598 parameter.
- 599 • Instances of CIM_Privilege shall be associated with the newly created instance of CIM_Role through
600 the CIM_MemberOfCollection association.
- 601 • The Cumulative Role Privilege of the newly associated instances of CIM_Privilege shall be equal to
602 the Cumulative Role Privilege of the embedded instances of CIM_Privilege contained in the
603 Privileges parameter.
- 604 • When the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities
605 instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of
606 FALSE, the CIM_Privilege instances shall be associated only with the newly created CIM_Role
607 instance and shall not be associated with any other instance of CIM_Role.

- 608 • When the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities
609 instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of
610 TRUE, the CIM_Privilege instances shall be associated with the newly created CIM_Role instance
611 and may be associated with any other instance of CIM_Role.
- 612 • Instances of CIM_RoleLimitedToTarget shall associate the newly created CIM_Role instance with
613 the instances referenced by the RoleLimitedToTargets parameter.
- 614 • Instances of CIM_ServiceAffectsElement shall associate the new CIM_Role instance and the
615 CIM_RoleBasedAuthorizationService instance.

616 When the properties of the embedded instances of RoleTemplate parameters and privileges are not fully
617 specified, the implementation may use its defaults to populate the resulting instances of CIM_Role and
618 CIM_Privilege.

619 The CreateRole() method shall return the value 2 (Error occurred) when the RoleCharacteristics property
620 of the RoleTemplate parameter's instance of CIM_Role contains the value 2 (Static).

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

624 No standard messages are defined for this method.

625 **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.

626 **Table 4 – CIM_RoleBasedAuthorizationService.CreateRole() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RoleTemplate	string	Embedded instance of CIM_Role that contains the non-key properties for the desired CIM_Role instance
IN, REQ	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, REQ	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

627 **8.1.1 CIM_RoleBasedAuthorizationService.CreateRole() Conditional Support**

628 When Authorized Role Management is supported and the SupportedMethods property array of the
629 Associated Role Management Capability of the instance of CIM_Role contains the value 4 (CreateRole),
630 the CreateRole() method shall be implemented and shall not return the value 1 (Not Supported).

631 When Authorized Role Management is not supported or the SupportedMethods property array of the
632 Associated Role Management Capability of the instance of CIM_Role does not contain the value
633 4 (CreateRole), the CreateRole() method shall not be implemented or shall always return the value 1 (Not
634 Supported).

635 **8.2 CIM_RoleBasedAuthorizationService.DeleteRole()**

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

637 The execution of the DeleteRole() method shall attempt to delete the CIM_Role instance referenced by
638 the Role parameter and the associated instances as described in this section.

639 When the CIM_Role instance referenced by the Role parameter is not associated with the
640 CIM_RoleBasedAuthorizationService instance through the CIM_ServiceAffectsElement association, the
641 DeleteRole() method shall fail and return the value 2 (Error occurred).

642 When the DeleteRole() method is implemented and the RoleCharacteristics property of the CIM_Role
643 instance referenced by the Role parameter contains a value of 2 (Static), the DeleteRole() method shall
644 fail and return the value 2 (Error occurred).

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

- 646 • All instances of the CIM_RoleLimitedToTarget association that reference the CIM_Role instance that
647 is referenced by the Role parameter shall be deleted.
- 648 • When the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities
649 instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of
650 FALSE, the implementation shall delete all the CIM_Privilege instances that are associated with the
651 CIM_Role instance that is referenced by the Role parameter.
- 652 • When the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities
653 instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of
654 TRUE, the implementation shall delete the CIM_Privilege instances that are only associated with the
655 CIM_Role instance that is referenced by the Role parameter.
- 656 • All instances of the CIM_MemberOfCollection association that reference the CIM_Role instance that
657 is referenced by the Role parameter shall be deleted.
- 658 • All instances of the CIM_OwningCollectionElement association that reference the CIM_Role instance
659 that is referenced by the Role parameter shall be deleted.
- 660 • The instance of the CIM_ServiceAffectsElement association that references the CIM_Role instance
661 that is referenced by the Role parameter and that references the
662 CIM_RoleBasedAuthorizationService instance shall be deleted.

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

666 No standard messages are defined for this method.

667 **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.

668 **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

669 **8.2.1 CIM_RoleBasedAuthorizationService.DeleteRole() Conditional Support**

670 When Authorized Role Management is supported and the SupportedMethods property array of the
 671 Associated Role Management Capability of the instance of CIM_Role contains the value 6 (DeleteRole),
 672 the DeleteRole() method shall be implemented and shall not return the value 1(Not Supported).

673 When Authorized Role Management is not supported or the SupportedMethods property array of the
 674 Associated Role Management Capability of the instance of CIM_Role does not contain the value
 675 6 (DeleteRole), the DeleteRole() method shall not be implemented or shall always return the value 1 (Not
 676 Supported).

677 **8.3 CIM_RoleBasedAuthorizationService.ModifyRole()**

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

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

- 680 • When the Privileges parameter is Null, the instances of CIM_Privilege that are associated with the
 681 Modified Role shall not be modified (see section 3.20).
- 682 • When the Privileges parameter is not Null and instances of CIM_Privilege are associated with the
 683 Modified Role through the CIM_MemberOfCollection association, the Cumulative Role Privilege of
 684 the associated instances of CIM_Privilege shall be equal to the Cumulative Role Privilege of the
 685 embedded instances of CIM_Privilege that are contained in the Privileges parameter.
- 686 • When the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities
 687 instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of
 688 FALSE, the CIM_Privilege instances shall be associated only with the Modified Role and shall not be
 689 associated with any other instance of CIM_Role.
- 690 • When the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities
 691 instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of
 692 TRUE, the CIM_Privilege instances shall be associated with the Modified Role and may be
 693 associated with any other instance of CIM_Role.
- 694 • An instance of CIM_RoleLimitedToTarget shall reference the Modified Role and an instance of
 695 CIM_ManagedElement only if a reference to the CIM_ManagedElement was contained in the
 696 RoleLimitedToTargets parameter.

697 The ModifyRole() method shall return the value 2 (Error occurred) when the Modified Role is not
 698 associated with the instance of CIM_RoleBasedAuthorizationService through an instance of
 699 CIM_ServiceAffectsElement.

700 The ModifyRole() method shall return the value 2 (Error occurred) when the Modified Role
 701 RoleCharacteristics property contains the value 2 (Static).

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

705 No standard messages are defined for this method.

706 **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.

707 **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

708 **8.3.1 CIM_RoleBasedAuthorizationService.ModifyRole() Conditional Support**

709 When Authorized Role Management is supported and the SupportedMethods property array of the
 710 Associated Role Management Capability of the instance of CIM_Role contains the value 5 (ModifyRole),
 711 the ModifyRole() method shall be implemented and shall not return the value 1(Not Supported).

712 When Authorized Role Management is not supported or the SupportedMethods property array of the
 713 Associated Role Management Capability of the instance of CIM_Role does not contain the value
 714 5 (ModifyRole), the ModifyRole() method shall not be implemented or shall always return the value 1 (Not
 715 Supported).

716 **8.4 CIM_RoleBasedAuthorizationService.AssignRoles()**

717 The AssignRoles() method is used to assign a security principal that is represented by an instance of
 718 CIM_Identity to zero or more roles represented by instances of CIM_Role.

719 When the CIM_Identity instance identified by the Identity parameter is not associated with an instance of
 720 CIM_AccountManagementService through the CIM_ServiceAffectsElement association, where the
 721 CIM_AccountManagementService is associated through the CIM_ServiceServiceDependency
 722 association with the instance of CIM_RoleBasedAuthorizationService upon which the method was
 723 invoked, the method shall return the value 2 (Failed).

724 When the Roles parameter contains a reference to an instance of CIM_Role that is not associated
 725 through the CIM_ServiceAffectsElement association with the instance of
 726 CIM_RoleBasedAuthorizationService upon which the method was invoked, the method shall return the
 727 value 2 (Failed).

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

731 No standard messages are defined for this method.

732 **Table 9 – CIM_RoleBasedAuthorizationService.AssignRoles() Method: Return Code Values**

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

733 **Table 10 – CIM_RoleBasedAuthorizationService.AssignRoles() Method: Parameters**

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

734 **8.4.1 CIM_RoleBasedAuthorizationService.AssignRoles() Conditional Support**

735 When Authorized Role Management is supported and the SupportedMethods property array of the
 736 Associated Role Management Capability of the instance of CIM_Role contains the value 6 (AssignRoles),
 737 the AssignRoles() method shall be implemented and shall not return the value 1 (Not Supported).

738 When Authorized Role Management is not supported or the SupportedMethods property array of the
 739 Associated Role Management Capability of the instance of CIM_Role does not contain the value
 740 6 (AssignRoles), the AssignRoles() method shall not be implemented or shall always return the value
 741 1 (Not Supported).

742 **8.5 CIM_RoleBasedAuthorizationService.ShowAccess()**

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

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

746 When the Subject parameter is not an instance of CIM_Identity, the method shall return the value 2
 747 (Failed).

748 When the CIM_Identity instance identified by the Identity parameter is not associated with an instance of
 749 CIM_AccountManagementService instance through the CIM_ServiceAffectsElement association, where
 750 the CIM_AccountManagementService is associated through the CIM_ServiceServiceDependency
 751 association with the instance of CIM_RoleBasedAuthorizationService upon which the method was
 752 invoked, the method shall return the value 2 (Failed).

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

- 755 • the set of instances of CIM_Role are those instances such that the instance of CIM_Identity specified
 756 by the Subject parameter is a member of the CIM_Role instance as defined in section 7.3.2
- 757 • the instance of CIM_ManagedElement specified by the Target parameter is in the scope of the
 758 CIM_Role instance as defined in section 7.1.1.1

759 • the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService
760 through the CIM_ServiceAffectsElement association

761 The OutSubjects and OutTargets parameters shall be Null when the method completes.

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

765 No standard messages are defined for this method.

766 **Table 11 – CIM_RoleBasedAuthorizationService.ShowAccess() Method: Return Code Values**

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

767 **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 EmbeddedObject	EmbeddedObject that contains the Cumulative Privilege

768 **8.5.1 CIM_RoleBasedAuthorizationService.ShowAccess() Conditional Support**

769 When Authorized Role Management is supported and the SupportedMethods property array of the
770 Associated Role Management Capability of the instance of CIM_Role contains the value 1 (ShowAccess),
771 the ShowAccess() method shall be implemented and shall not return the value 1(Not Supported).

772 When Authorized Role Management is not supported or the SupportedMethods property array of the
773 Associated Role Management Capability of the instance of CIM_Role does not contain the value
774 1 (ShowAccess), the ShowAccess() method shall not be implemented or shall always return the value
775 1 (Not Supported).

776 **8.6 CIM_RoleBasedAuthorizationService.ShowRoles()**

777 The ShowRoles() method is used to assign a security principal that is represented by an instance of
778 CIM_Identity to zero or more roles represented by instances of CIM_Role.

779 When the Subject parameter is not an instance of CIM_Identity, the method shall return the value 2
780 (Failed).

781 When the Subject parameter is not Null and the CIM_Identity instance identified by the Subject parameter
782 is not associated with an instance of CIM_AccountManagementService through the
783 CIM_ServiceAffectsElement association, where the CIM_AccountManagementService is associated
784 through the CIM_ServiceServiceDependency association with the instance of
785 CIM_RoleBasedAuthorizationService upon which the method was invoked, the method shall return the
786 value 2 (Failed).

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

788 When the Subject and Target parameters are not Null, upon successful completion of the method
 789 • the Roles parameter shall contain an embedded instance of CIM_Role for each instance of
 790 CIM_Role such that the instance of CIM_Identity specified by the Subject parameter is a member of
 791 the CIM_Role instance as defined in section 7.3.2
 792 • the instance of CIM_ManagedElement specified by the Target parameter is in the scope of the
 793 CIM_Role instance as defined in section 7.1.1.1
 794 • the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService
 795 through the CIM_ServiceAffectsElement association

796 When the Subject parameter is not Null and the Target parameter is Null, upon successful completion of
 797 the method
 798 • the Roles parameter shall contain an embedded instance of CIM_Role for each of instance of
 799 CIM_Role such that the instance of CIM_Identity specified by the Subject parameter is a member of
 800 the CIM_Role instance as defined in section 7.3.2
 801 • the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService
 802 through the CIM_ServiceAffectsElement association

803 When the Subject parameter is Null and the Target parameter is not Null, upon successful completion of
 804 the method
 805 • the Roles parameter shall contain an embedded instance of CIM_Role for each of instance of
 806 CIM_Role such that the instance of CIM_ManagedElement specified by the Target parameter is in
 807 the scope of the CIM_Role instance as defined in section 7.1.1.1
 808 • the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService
 809 through the CIM_ServiceAffectsElement association

810 When the Subject and Target parameters are both Null, upon successful completion of the method, the
 811 Roles parameter shall contain an embedded instance of CIM_Role for each of instance of CIM_Role such
 812 that the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService
 813 through the CIM_ServiceAffectsElement association.

814 For each instance of CIM_Role for which the Roles parameter contains an embedded instance of
 815 CIM_Role, the Privileges parameter shall contain at the same array index an embedded instance of
 816 CIM_Privilege that represents the Cumulative Privilege of the CIM_Role as defined in section 7.1.3.2.

817 The ShowRoles() method’s return code values shall be as specified in Table 13 where the method
 818 execution behavior matches the return code description. The ShowRoles() method’s parameters are
 819 specified in Table 14.

820 No standard messages are defined for this method.

821 **Table 13 – CIM_RoleBasedAuthorizationService.ShowRoles() Method: Return Code Values**

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

822

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	Reference to the CIM_ManagedElement instance
OUT	Roles	string EmbeddedInstance (CIM_Role)	Array of embedded instances of CIM_Role
OUT	Privileges	string EmbeddedInstance (CIM_Privilege)	Array of embedded instances of CIM_Privilege

823 **8.6.1 CIM_RoleBasedAuthorizationService.ShowRoles() Conditional Support**

824 When Authorized Role Management is supported and the SupportedMethods property array of the
 825 Associated Role Management Capability of the instance of CIM_Role contains the value 7 (ShowRoles),
 826 the ShowRoles() method shall be implemented and shall not return the value 1 (Not Supported).

827 When Authorized Role Management is not supported or the SupportedMethods property array of the
 828 Associated Role Management Capability of the instance of CIM_Role does not contain the value
 829 7 (ShowRoles), the ShowRoles() method shall not be implemented or shall always return the value 1 (Not
 830 Supported).

831 **8.7 Profile Conventions for Operations**

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

837 The default list of operations is as follows:

- 838 • GetInstance
- 839 • EnumerateInstances
- 840 • EnumerateInstanceNames
- 841 • Associators
- 842 • AssociatorNames
- 843 • References
- 844 • ReferenceNames

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

847 **8.8 CIM_ConcreteDependency**

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

850 **Table 15 – Operations: CIM_ConcreteDependency**

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

851 **8.9 CIM_ElementCapabilities**

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

854 **Table 16 – Operations: CIM_ElementCapabilities**

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

855 **8.10 CIM_HostedService**

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

858 **Table 17 – Operations: CIM_HostedService**

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

859 **8.11 CIM_MemberOfCollection**

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

862 **Table 18 – Operations: CIM_MemberOfCollection**

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

863 **8.12 CIM_OwningCollectionElement**

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

866 **Table 19 – Operations: CIM_OwningCollectionElement**

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

867 **8.13 CIM_Privilege**

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

870 **Table 20 – Operations: CIM_Privilege**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.13.1.	None

871 **8.13.1 CIM_Privilege—ModifyInstance**

872 When Authorized Role Management is not supported or the SupportedMethods property array of the
873 Associated Role Management Capability of the instance of CIM_Role does not contain the value
874 8 (ModifyPrivilege), the ModifyInstance operation shall not be supported.

875 When Authorized Role Management is supported and the SupportedMethods property array of the
 876 Associated Role Management Capability of the instance of CIM_Privilege contains the value
 877 8 (ModifyPrivilege), the ModifyInstance operation shall be supported except as follows:

- 878 • The ModifyInstance operation shall not be supported on the Granted Privileges or Denied Privileges
 879 that are associated with an instance of CIM_Role when the CIM_Role.RoleCharacteristics property
 880 contains the value 2 (Static).
- 881 • The ModifyInstance operation shall not be supported on the Template Privileges.

882 **8.14 CIM_RoleBasedManagementCapabilities**

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

884 **8.15 CIM_Role**

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

886 **8.16 CIM_RoleBasedAuthorizationService**

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

888 **8.17 CIM_RoleLimitedToTarget**

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

891 **Table 21 – Operations: CIM_RoleLimitedToTarget**

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

892 **8.18 CIM_ServiceAffectsElement**

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

895 **Table 22 – Operations: CIM_ServiceAffectsElement**

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

896 **8.19 CIM_ServiceServiceDependency**

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

899 **Table 23 – Operations: CIM_ServiceServiceDependency**

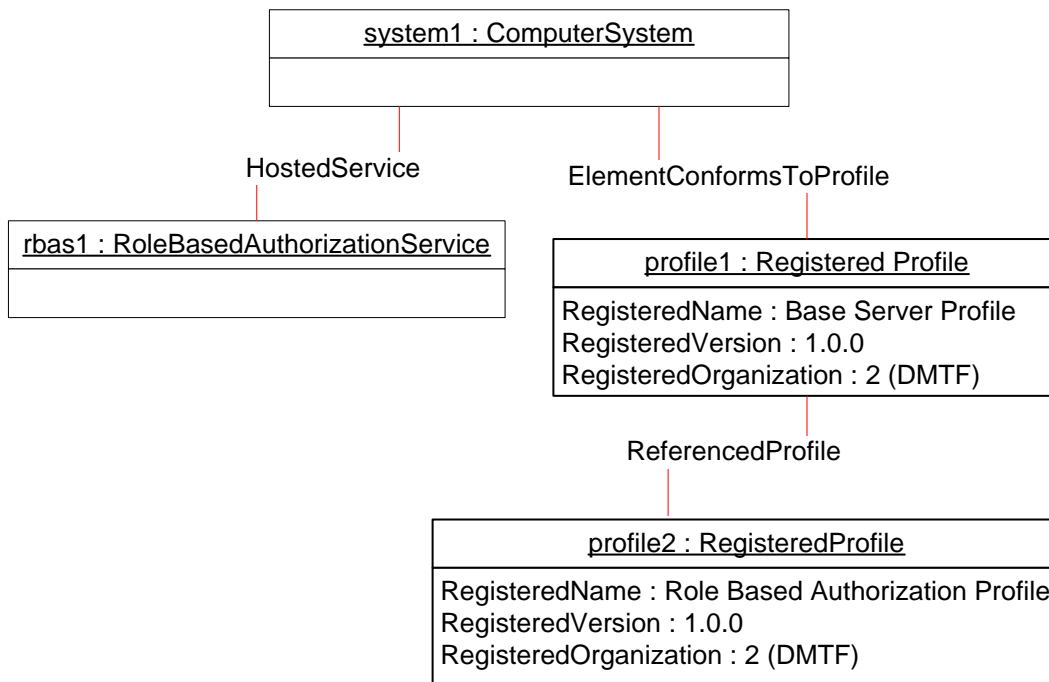
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None
EnumerateInstances	Unspecified	None
EnumerateInstanceNames	Unspecified	None

900 **9 Use Cases**

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

904 **9.1 Profile Registration**

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



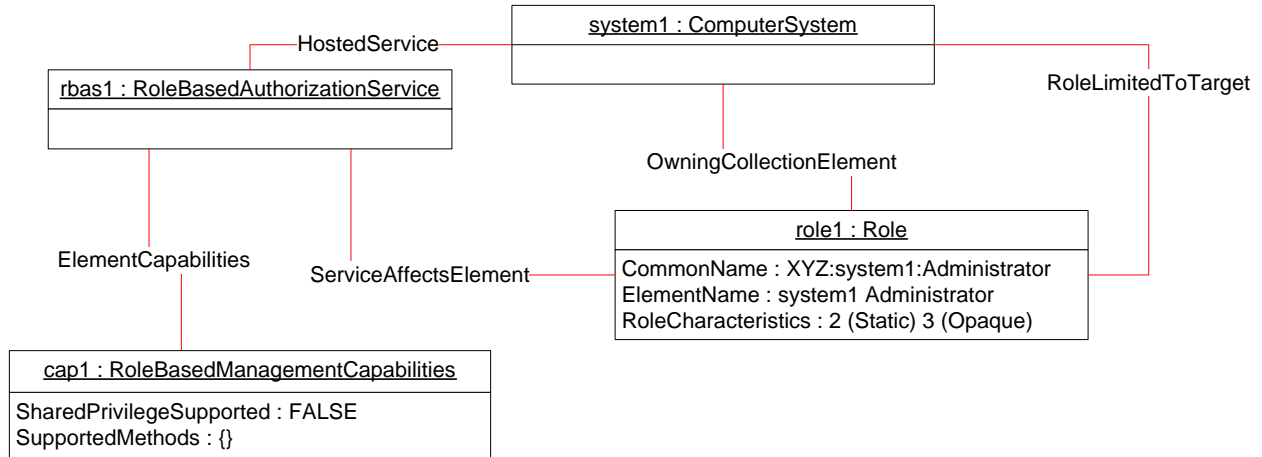
908

909

Figure 2 – Profile Registration

910 **9.2 Minimal Instantiation of the Profile**

911 Figure 3 describes a possible minimal instantiation of the *Role Based Authorization Profile*. In this
 912 instantiation, role1 is described as being a system1 administrator role. The scope of role1 is limited to
 913 system1 as shown by the instance of the CIM_RoleLimitedToTarget association. role1 is opaque and
 914 static. The rights granted by the role are not explicitly modeled. No methods are supported for
 915 management of the role, which is indicated by the empty array for the SupportedMethods property of
 916 cap1.

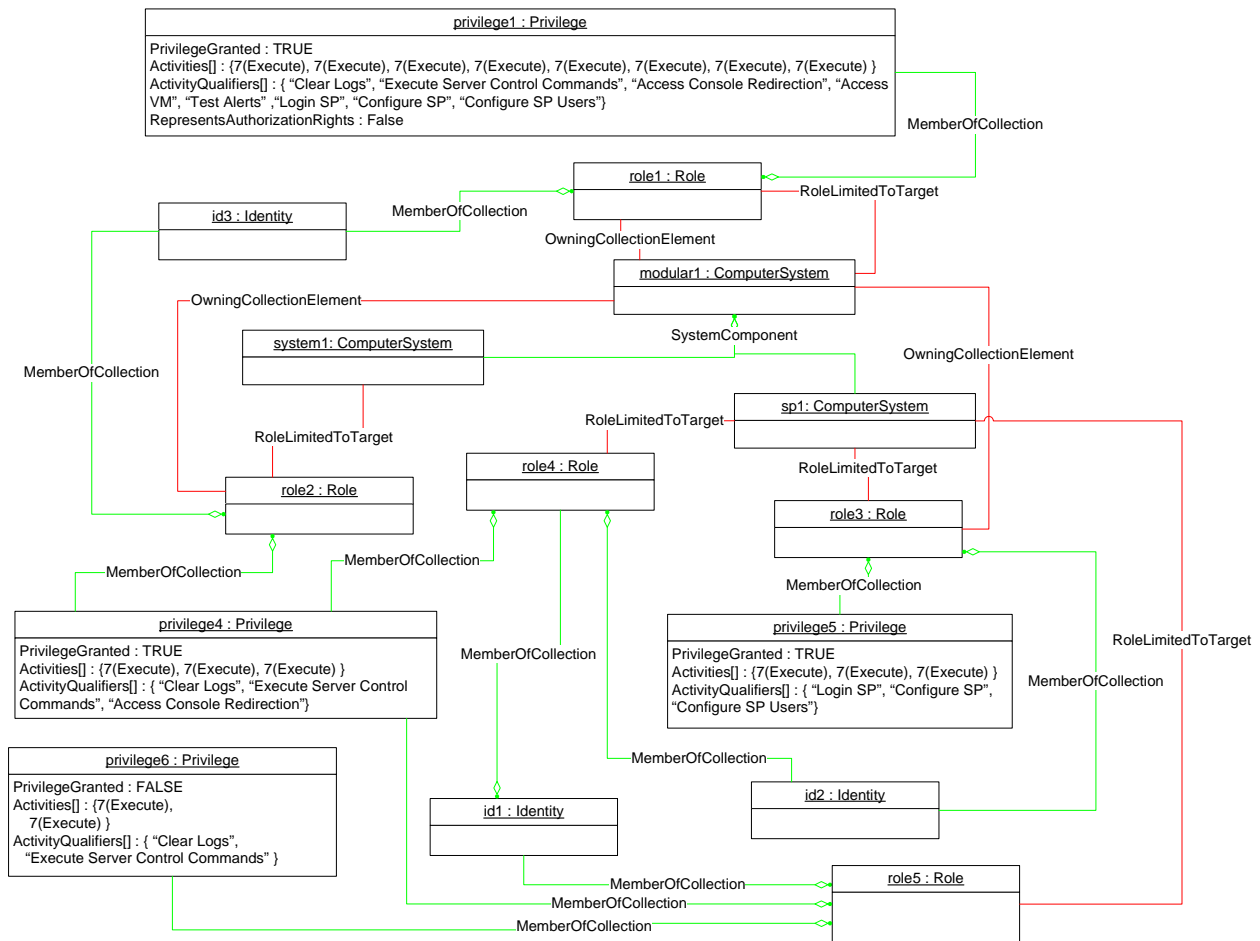


917

918 **Figure 3 – Minimal Instantiation**

919 **9.3 Evaluating Scope and Privileges**

920 Figure 4 illustrates the behavior of the `CIM_PrivilegeManagementService.ShowAccess()` and
 921 `CIM_PrivilegeManagementService.ShowRoles()` methods. The diagram illustrates two systems (system1
 922 and sp1) contained within a third system (modular1). role1 is explicitly scoped to modular1; system1 and
 923 sp1 are within modular1, so they are also within the scope of role1. role2 is explicitly scoped to system1.
 924 role3, role4, and role5 are explicitly scoped to sp1.



925

926

Figure 4 – Cumulative Role Privilege Example

927 **9.3.1 CIM_PrivilegeManagementService.ShowRoles()**

928 Given a value of id1 for the Subject parameter and Null for the Target parameter, the ShowRoles()
 929 method will return information about each instance of CIM_Role of which id1 is a member. Thus two
 930 embedded instances of CIM_Role will be in the Roles parameter, one corresponding to role5 and one
 931 corresponding to role4. Two embedded instances of CIM_Privilege will be returned in the Privileges
 932 parameter, one reflecting the cumulative privileges of role5 and the other those of role4.

933 The embedded instance of CIM_Privilege that corresponds to the Cumulative Privilege of role4 is
 934 constructed by adding the Granted Privileges (privilege4) to the Cumulative Privilege and subtracting from
 935 the Cumulative Privilege the intersection with the Denied Privilege (privilege6). This results in the
 936 following values for the Activities and ActivityQualifier properties:

- 937
- CIM_Privilege.Activities = { 7(Execute) }
 - 938 • CIM_Privilege.ActivityQualifiers = { "Access Console Redirection" }

939 **9.3.2 CIM_PrivilegeManagementService.ShowAccess()**

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

942 **9.3.2.1 Example: CIM_PrivilegeManagementService.ShowAccess()**

943 Subject = id1

944 Target = modular1

945 CIM_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute) }

946 CIM_Privilege.ActivityQualifiers = { "Access Console Redirection", "Login SP", "Configure SP", "Configure
947 SP Users" }

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

952 **9.3.2.2 Example: CIM_PrivilegeManagementService.ShowAccess()**

953 Subject = id3

954 Target = modular1

955 CIM_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),
956 7(Execute), 7(Execute) }

957 CIM_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console
958 Redirection", "Access VM", "Test Alerts", "Login SP", "Configure SP", "Configure SP Users" }

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

962 **9.3.2.3 Example: CIM_PrivilegeManagementService.ShowAccess()**

963 Subject = id3

964 Target = system1

965 CIM_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),
966 7(Execute), 7(Execute) }

967 CIM_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console
968 Redirection", "Access VM", "Test Alerts", "Login SP", "Configure SP", "Configure SP Users" }

969 id3 belongs to role1 and role2. system1 is contained in modular1 and modular1 is in the scope of role1.
970 Therefore, sp1 is in the scope of role1. system1 is explicitly within the scope of role2. The intersection of
971 the roles is role1 and role2. Therefore, the set of roles to which the algorithm in section 7.1.3.3 will be
972 applied consists of role1 and role2. The Cumulative Privilege of role1 is a superset of the Cumulative
973 Privilege of role2. Therefore. the out parameter contains the Cumulative Privilege of role1.

974 **9.3.2.4 Example: CIM_PrivilegeManagementService.ShowAccess()**

975 Subject = id3

976 Target = sp1

977 CIM_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute),
978 7(Execute), 7(Execute) }

979 CIM_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console
980 Redirection", "Access VM", "Test Alerts", "Login SP", "Configure SP", "Configure SP Users"}

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

985 **9.3.2.5 Example: CIM_PrivilegeManagementService.ShowAccess()**

986 Subject = id2

987 Target = sp1

988 CIM_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute) }

989 CIM_Privilege.ActivityQualifiers = { "Clear Logs", "Execute Server Control Commands", "Access Console
990 Redirection", "Login SP", "Configure SP", "Configure SP Users"}

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

995 **9.4 Scope of the Role and Privileges for a Managed Element**

996 Figure 5 shows a system that has three local accounts and uses role membership to manage the
 997 privileges for a user account. This system has three local accounts: acct1, acct2, and acct3. acct1 and
 998 acct2 currently have the privileges of role1. acct3 does not have any privileges.

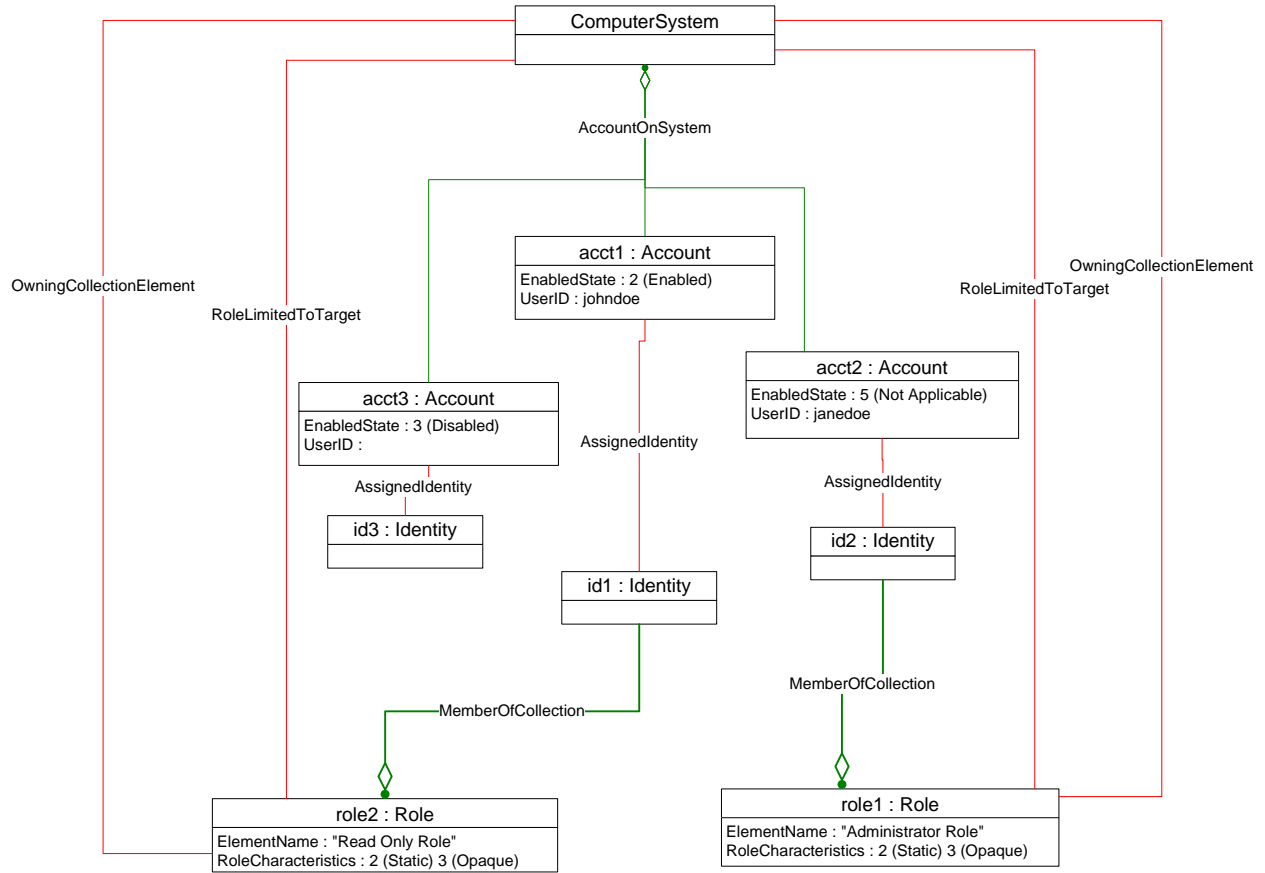
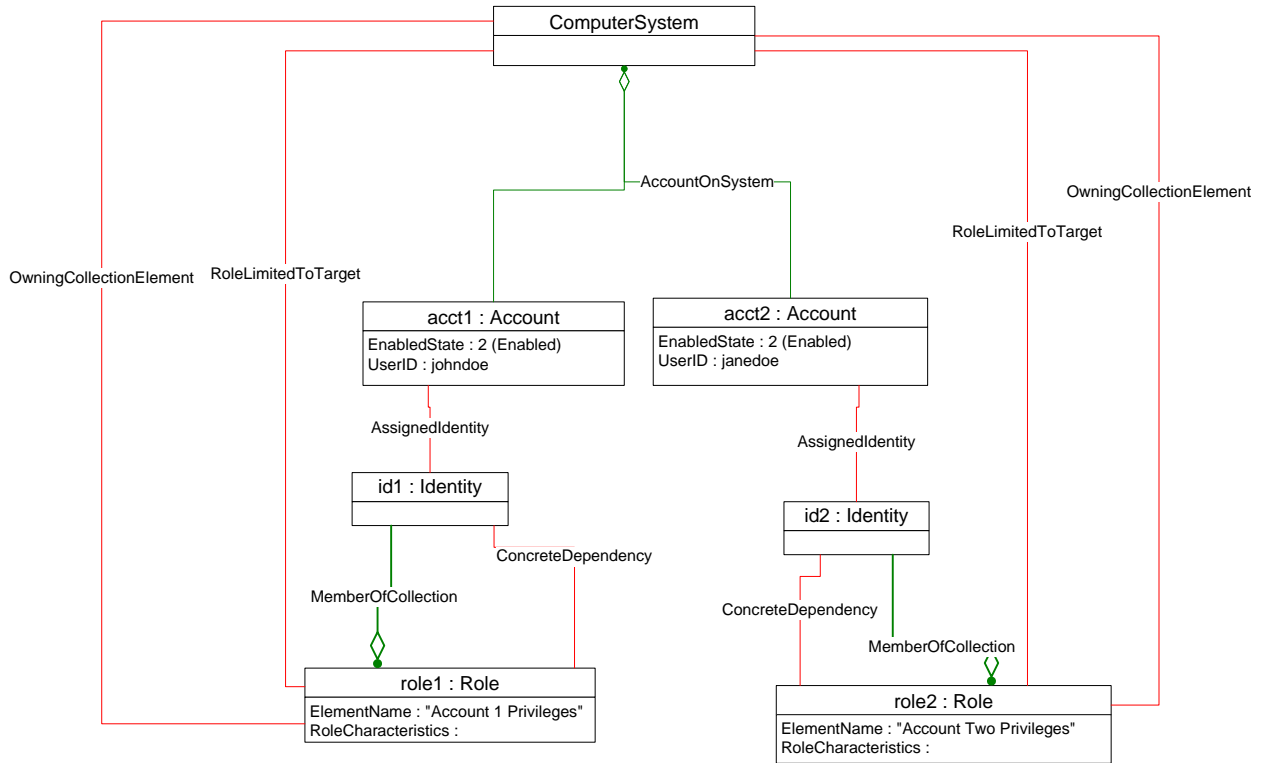


Figure 5 – Roles and Privileges for Principals

999

1000

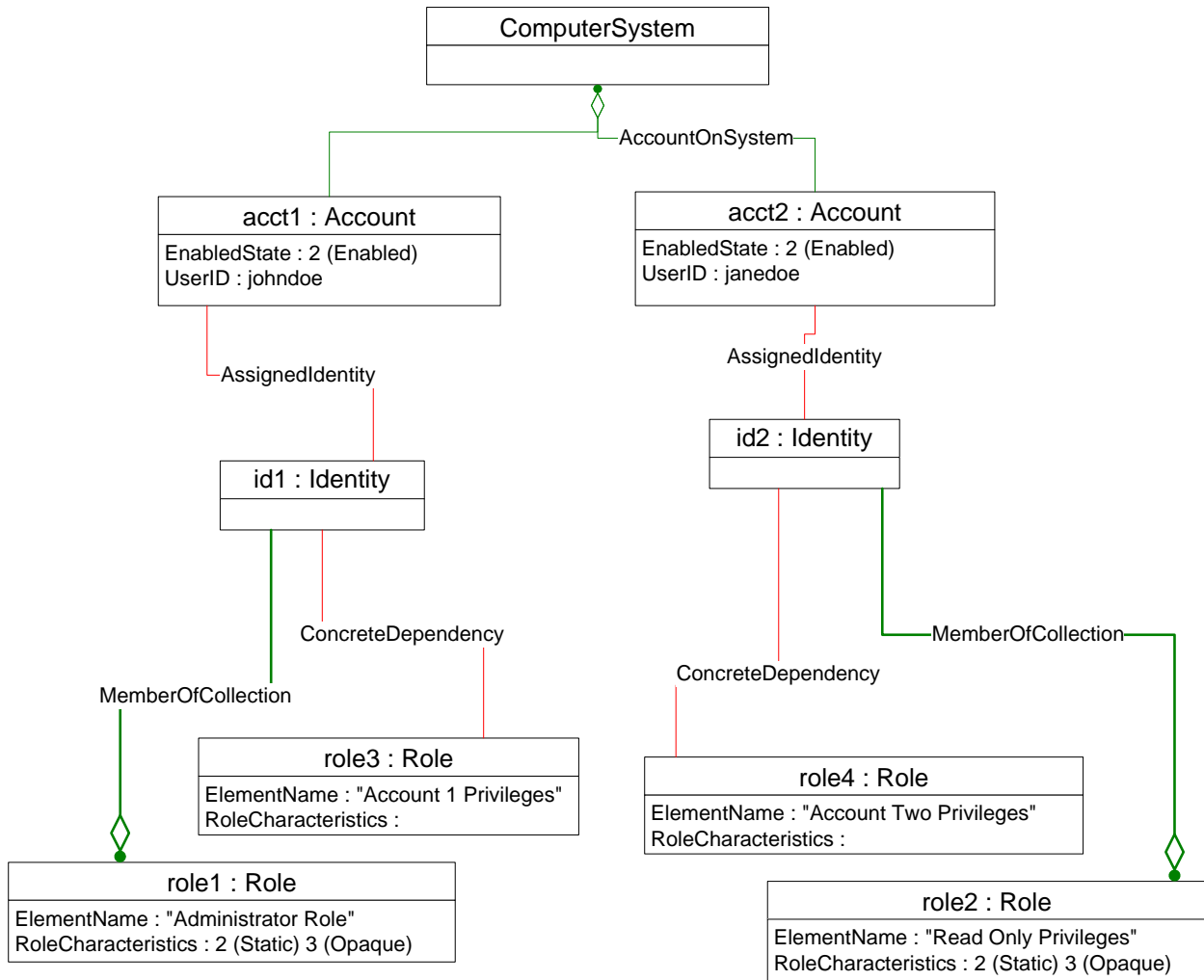
1001 Figure 6 shows a system that has two local accounts and manages privileges for individual accounts.
 1002 This system has two local accounts: acct1 and acct2. Privileges for acct1 and acct2 are managed through
 1003 role1 and role2, respectively, as indicated by the CIM_ConcreteDependency associations. No common
 1004 roles are defined; therefore, privileges for each account can be managed only through their respective
 1005 dedicated roles.



1006
 1007

Figure 6 – Fixed Accounts with Role Membership Privilege Management

1008 Figure 7 shows a system that has two local accounts. Privileges for the accounts are managed either
 1009 through assignment to a pre-defined role (role1 and role2) or through modification of privileges granted to
 1010 a dedicated role (role3 and role4).



1011

1012

Figure 7 – Fixed Accounts with Individual Account Privilege Management

1013 **9.5 Service Processor Roles Use Cases**

1014 This section provides object diagrams for a possible implementation of authorized roles for a service
 1015 processor.

1016 Figure 8 **Error! Reference source not found.** represents a possible instantiation of the *Role Based*
 1017 *Authorization Profile* for IPMI-based service processor roles. Three roles are represented: role1, role2,
 1018 role3. These roles have the scope that includes system1 and the service processor, sp1. The privileges
 1019 for the authorized roles are represented through the IPMI commands that each role allows the associated
 1020 user to execute. The security principals id1, id2, and id3, are each associated with Serial1, protoendpt2,
 1021 and protoendpt2, respectively, representing the communication channel that has handled the
 1022 authentication. id1, id2, and id3 have privileges to act within system1 as denoted by the instances of
 1023 CIM_RoleLimitedToTarget that associate their member roles to system1. Because sp1 is a component of
 1024 system1, id1, id2, and id3 have the same privileges within sp1.

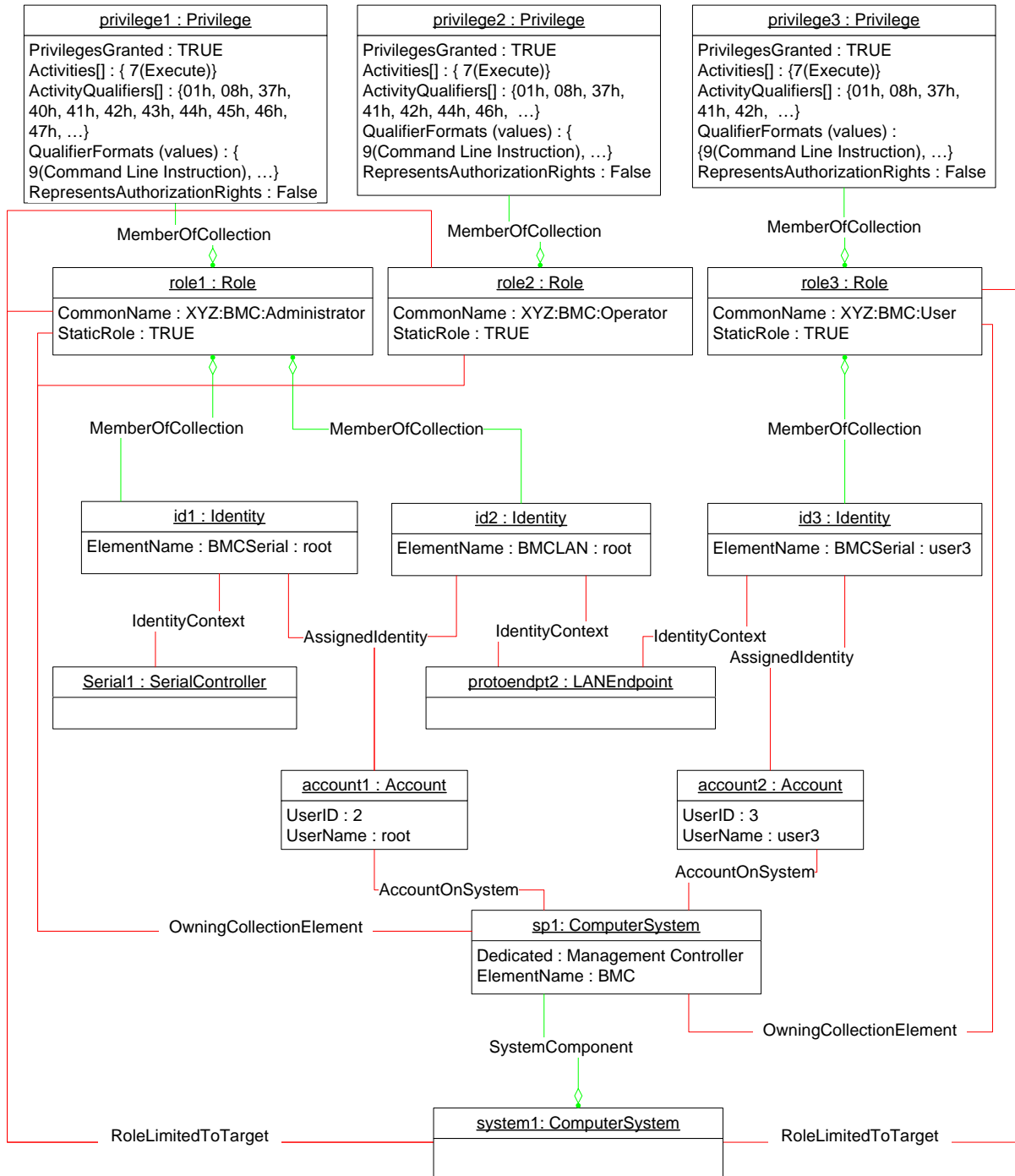
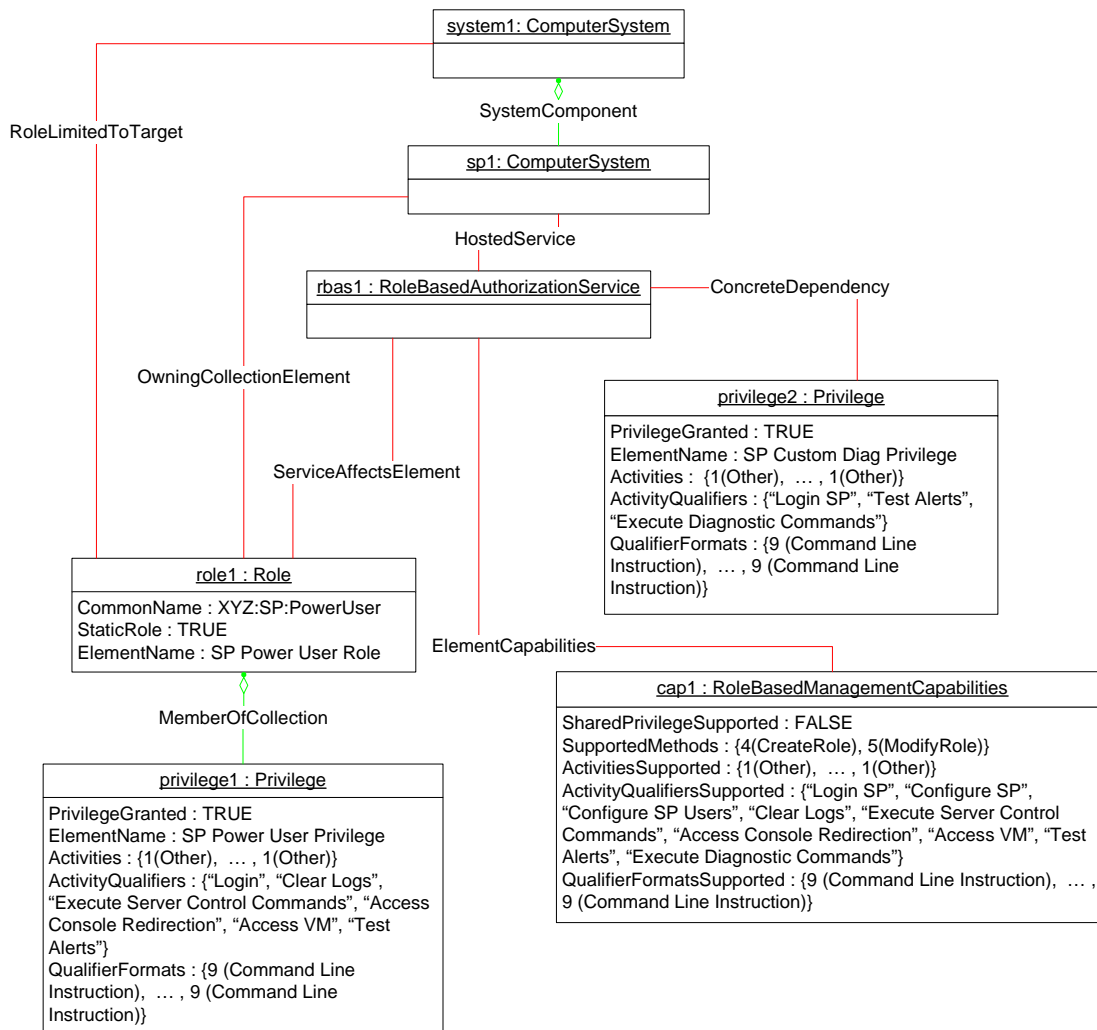


Figure 8 – IPMI Service Processor with Role Management

1028 Figure 8 represents another instantiation of the *Role Based Authorization Profile* for service processor
 1029 roles. system1 hosts sp1, which represents the service processor. sp1 has has a predefined role, role1,
 1030 scope extends to the host computer system, system1, and the service processor itself, sp1. role1's
 1031 privileges are represented by privilege1. cap1 advertises the capabilities for the client to do Authorized
 1032 Role Management. cap1's SupportedMethods property contains two values: 4 (CreateRole) and 5
 1033 (ModifyRole), which advertises to the client that Authorized Role Management is supported with
 1034 CreateRole() and ModifyRole() extrinsic methods.

1035 To execute the CreateRole() method successfully, the client needs to know the type of privileges that the
 1036 new role can support. Because the underlying device has binary representation of activities, the
 1037 implementation has populated the ActivitiesSupported, ActivityQualifiersSupported, and
 1038 QualifierFormatsSupported properties of cap1, and the instrumentation has instantiated a Template
 1039 Privilege, privilege2, to give the client further guidance on the construction of the Privileges parameter of
 1040 the CreateRole() method of rbas1.



1041

1042

Figure 9 – IPMI Service Processor with Role Management

1043 **9.6 Determine the Roles Managed by a Service**

1044 Given an instance of `CIM_RoleBasedAuthorizationService`, a client can determine the instances of
1045 `CIM_Role` managed by the instance of `CIM_RoleBasedAuthorizationService` as follows:

- 1046 1) Find the instance of `CIM_RoleBasedManagementCapabilities` that is associated with the target
1047 instance through an instance of `CIM_ElementCapabilities`.
- 1048 2) If the `CIM_RoleBasedManagementCapabilities.SupportedMethods` property contains the value 7
1049 (`ShowRoles`), invoke the `CIM_RoleBasedAuthorizationService.ShowRoles()` method, specifying `Null`
1050 for the `Subject` and `Target` parameters.

1051 Upon successful completion, the `Roles` parameter will contain an embedded instance of `CIM_Role`
1052 for each `CIM_Role` instance managed by the service.

- 1053 3) If the `CIM_RoleBasedManagementCapabilities.SupportedMethods` property does not contain the
1054 value 7 (`ShowRoles`), find the instances of `CIM_Role` that are associated through the
1055 `CIM_ServiceAffectsElement` association.

1056 **9.7 Determine Candidate Roles for a Security Principal**

1057 Given an instance of `CIM_Identity` that represents a security principal, a client can determine all of the
1058 instances of `CIM_Role` to which the `CIM_Identity` instance could be assigned as follows:

- 1059 1) Find the instance of `CIM_IdentityManagementService` that is associated with the `CIM_Identity`
1060 instance through the `CIM_ServiceAffectsElement` association.
- 1061 2) Find the instances of `CIM_RoleBasedAuthorizationService` that are associated with the
1062 `CIM_IdentityManagementService` through the `CIM_ServiceServiceDependency` association.
- 1063 3) For each instance of `CIM_RoleBasedAuthorizationService`, use the steps in section 9.6 to find the
1064 instances of `CIM_Role` that are managed by the service.

1065 The union of the instances of `CIM_Role` from step 3) form the set of instances of `CIM_Role` to which
1066 the `CIM_Identity` instance could be assigned.

1067 **9.8 Determine the Roles to Which a Security Principal Is Currently Assigned**

1068 Given an instance of `CIM_Identity` that represents a security principal, a client can determine the
1069 instances of `CIM_Role` to which the `CIM_Identity` instance is currently assigned as follows:

- 1070 1) Find the instance of `CIM_IdentityManagementService` that is associated with the `CIM_Identity`
1071 instance through the `CIM_ServiceAffectsElement` association.
- 1072 2) Find the instances of `CIM_RoleBasedAuthorizationService` that are associated with the
1073 `CIM_IdentityManagementService` through the `CIM_ServiceServiceDependency` association.
- 1074 3) For each instance of `CIM_RoleBasedAuthorizationService`, find the instance of
1075 `CIM_RoleBasedManagementCapabilities` that is associated through the `CIM_ElementCapabilities`
1076 association.
- 1077 4) If the `CIM_RoleBasedManagementCapabilities.SupportedMethods` property contains the value
1078 7 (`ShowRoles`), invoke the `CIM_RoleBasedAuthorizationService.ShowRoles()` method, specifying a
1079 reference to the `CIM_Identity` instance as the value of the `Subject` parameter and `Null` for the `Target`
1080 parameter.

1081 Upon successful completion, the `Roles` parameter will contain an embedded instance of `CIM_Role`
1082 for each `CIM_Role` instance managed by the service.

- 1083 5) If the `CIM_RoleBasedManagementCapabilities.SupportedMethods` property does not contain the
1084 value 7 (`ShowRoles`), find all of the instances of `CIM_Role` that are associated with the `CIM_Identity`
1085 instance through the `CIM_MemberOfCollection` association.

1086 **9.9 Determine the Roles that Scope a Managed Element**

1087 Given an instance of CIM_ManagedElement, a client can determine the instances of CIM_Role that
1088 scope the target instance as follows:

- 1089 1) Find all instances of CIM_RoleBasedAuthorizationService.
- 1090 2) For each instrumented instance of CIM_RoleBasedAuthorizationService, find the instance of
1091 CIM_RoleBasedManagementCapabilities that is associated through the CIM_ElementCapabilities
1092 association.
- 1093 3) If the CIM_RoleBasedManagementCapabilities.SupportedMethods property contains the value
1094 7 (ShowRoles), invoke the ShowRoles() method, specifying Null for the Subject parameter and a
1095 reference to the CIM_ManagedElement instance as the value of the Target parameter.

1096 **9.10 Determine the Current Privileges of a Security Principal for a Managed 1097 Element**

1098 Given an instance of CIM_Identity that represents a security principal and an instance of
1099 CIM_ManagedElement, a client can determine the current privileges of the CIM_Identity instance for
1100 managing the instance of CIM_ManagedElement as follows:

- 1101 1) Find the instance of CIM_IdentityManagementService that is associated with the CIM_Identity
1102 instance through the CIM_ServiceAffectsElement association.
- 1103 2) Find the instances of CIM_RoleBasedAuthorizationService that are associated with the
1104 CIM_IdentityManagementService through the CIM_ServiceServiceDependency association.
- 1105 3) For each instance of CIM_RoleBasedAuthorizationService, find the instance of
1106 CIM_RoleBasedManagementCapabilities that is associated through the CIM_ElementCapabilities
1107 association.
- 1108 4) If the CIM_RoleBasedManagementCapabilities.SupportedMethods property contains the value
1109 1 (ShowAccess), invoke the CIM_RoleBasedAuthorizationService.ShowAccess() method, specifying
1110 a reference to the CIM_Identity instance as the value of the Subject parameter and a reference to
1111 the instance of CIM_ManagedElement for the Target parameter.

1112 Upon successful completion, the Privileges parameter will contain an embedded instance of
1113 CIM_Privilege that represents the Cumulative Privilege granted to the security principal by the
1114 instances of CIM_Role that are managed by the instance of CIM_RoleBasedAuthorizationService.

1115 **9.11 Modify the Privileges of an Existing Role**

1116 A client can modify the privileges of an existing role as follows:

- 1117 1) If the SupportedMethods property of the Associated Role Management Capability of the selected
1118 CIM_Role instance has a value of 8 (ModifyPrivilege), and if the RoleCharacteristics property of the
1119 selected instance of CIM_Role does not have the value 2 (Static), then select the CIM_Privilege
1120 instances associated with the instance of CIM_Role.
- 1121 2) Execute the ModifyInstance operation on the selected instances of CIM_Privilege, modifying the
1122 privilege accordingly.

1123 Otherwise, the role is static and its privileges cannot be modified.

1124 **9.12 Create a New Role**

1125 A client can create a new role as follows:

- 1126 1) If the SupportedMethods property of the associated CIM_PrivilegeManagementCapabilities instance
1127 has a value of 4 (CreateRole), select the instance of CIM_RoleBasedAuthorizationService that is

- 1128 associated with the given instance of CIM_ComputerSystem that represents the service processor
1129 through an instance of CIM_HostedService.
- 1130 2) Select the instance of CIM_RoleBasedManagementCapabilities that is associated with the selected
1131 instance of CIM_RoleBasedAuthorizationService.
- 1132 3) Construct the parameters for the CIM_RoleBasedAuthorizationService.CreateRole() method in the
1133 following way:
- 1134 • RoleTemplate: Construct the desired embedded instance of CIM_Role.
 - 1135 • OwningSystem: Construct the CIM reference to the instance of CIM_ComputerSystem that
1136 represents the service processor.
 - 1137 • Privileges: Construct the embedded instance of CIM_Privilege based on the
1138 ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties of
1139 the selected instance of CIM_RoleBasedManagementCapabilities, or based on the Template
1140 Privilege associated with the CIM_RoleBasedAuthorizationService instance.
 - 1141 • RoleLimitedToTargets: Construct the CIM reference to the instance of CIM_ComputerSystem
1142 that is associated with the instance of CIM_ComputerSystem that represents the service
1143 processor through an instance of CIM_SystemComponent.
- 1144 4) Execute the CIM_RoleBasedAuthorizationService.CreateRole() method with the preceding
1145 parameters.

1146 **9.13 Determine Whether Privilege Management Is Supported for a Principal**

1147 A client can determine whether privilege management is supported for a security principal as follows:

- 1148 1) Starting at the instance of CIM_Identity that represents the security principal, find the instances of
1149 CIM_AccountManagementService that are associated through the CIM_ServiceAffectsElement
1150 association.
- 1151 2) For each instance of CIM_AccountManagementService, determine if at least one instance of
1152 CIM_RoleBasedAuthorizationService is associated through the CIM_ServiceServiceDependency
1153 association.
- 1154 3) If at least one instance of CIM_RoleBasedAuthorizationService is associated with at least one
1155 instance of CIM_AccountManagementService, privilege management is supported for the security
1156 principal.

1157 **9.14 Determine Whether One-to-One Privilege Management Is Supported for an 1158 Account**

1159 A client can determine whether authorization for a security principal can be managed using one-to-one
1160 correspondence as follows:

1161 Starting at the target instance of CIM_Identity, query for an instance of CIM_ConcreteDependency that
1162 references the CIM_Identity instance and an instance of CIM_Role.

1163 If an instance exists, authorization for the CIM_Account can be managed through one-to-one
1164 correspondence. Note that authorization through role membership could also be supported.

1165 **9.15 Assign Custom Privileges to an Identity**

1166 A client can assign custom privileges to an instance of CIM_Account as follows:

- 1167 1) Determine whether privileges for the CIM_Account are managed through one-to-one
1168 correspondence or role membership as described in section 9.14.

- 1169 If privileges are not managed through one-to-one correspondence, it is necessary to create a custom
 1170 role that has the desired privileges. See section 9.12 for information about how to create a role with
 1171 the desired privileges.
- 1172 2) If privileges are managed through one-to-one correspondence, find the instance of CIM_Identity that
 1173 is associated with the CIM_Account instance.
- 1174 3) Find the instance of CIM_Role that is associated with the CIM_Identity instance through an instance
 1175 of CIM_ConcreteDependency.
- 1176 4) If the CIM_Identity instance is not already associated with the instance of CIM_Role from step 3)
 1177 through an instance of CIM_MemberOfCollection, use CreateInstance to create an instance of
 1178 CIM_MemberOfCollection that associates the CIM_Identity instance with the CIM_Role instance.
- 1179 5) If the CIM_Identity is associated with the instance of CIM_Role other than that from step 3) through
 1180 an instance of CIM_MemberOfCollection, use DeleteInstance to delete the instance of
 1181 CIM_MemberOfCollection that associates the CIM_Identity instance with the CIM_Role instance.
- 1182 6) Perform role modification on the instance of CIM_Role from step 3) as specified in section 9.6.

1183 **10 CIM Elements**

1184 Table 24 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 1185 implemented as described in Table 24. Sections 7 (“Implementation”) and 8 (“Methods”) may impose
 1186 additional requirements on these elements.

1187 **Table 24 – CIM Elements: Role Based Authorization Profile**

Element Name	Requirement	Description
Classes		
CIM_ConcreteDependency (Privilege)	Optional	See section 10.1.
CIM_ConcreteDependency (Role)	Optional	See section 10.2.
CIM_ElementCapabilities	Mandatory	See sections 10.3 and 7.2.1.
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 10.9 and 7.2.2.
CIM_RegisteredProfile	Mandatory	See section 10.10.
CIM_Role	Mandatory	See section 10.11.
CIM_RoleBasedAuthorizationService	Mandatory	See section 7.2 and 10.12.
CIM_RoleLimitedToTarget	Mandatory	See section 10.13.
CIM_ServiceAffectsElement	Mandatory	See section 10.14.
CIM_ServiceServiceDependency	Optional	See section 10.15.
Indications		
None defined in this profile		

1188 **10.1 CIM_ConcreteDependency (Privilege)**

1189 CIM_ConcreteDependency is used to associate a Template Privilege with an instance of
 1190 CIM_RoleBasedAuthorizationService. Table 25 contains the requirements for elements of this class.

1191 **Table 25 – Class: CIM_ConcreteDependency (Privilege)**

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

1192 **10.2 CIM_ConcreteDependency (Role)**

1193 CIM_ConcreteDependency is used to associate an instance of CIM_Identity with an instance of
 1194 CIM_Role. Table 26 contains the requirements for elements of this class.

1195 **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

1196 **10.3 CIM_ElementCapabilities**

1197 CIM_ElementCapabilities is used to associate an instance of CIM_RoleBasedAuthorizationService with
 1198 an instance of CIM_RoleBasedManagementCapabilities that describes the capabilities of the role
 1199 management. Table 27 contains the requirements for elements of this class.

1200 **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.

1201 **10.4 CIM_HostedService**

1202 CIM_HostedService is used to associate an instance of CIM_RoleBasedAuthorizationService with an
 1203 instance of CIM_ComputerSystem that is the computer system hosting the service. Table 28 contains the
 1204 requirements for elements of this class.

1205 **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..*

1206 **10.5 CIM_MemberOfCollection (Privilege)**

1207 CIM_MemberOfCollection is used to associate an instance of CIM_Privilege with an instance of
 1208 CIM_Role that represents the role that contains the privilege. Table 29 contains the requirements for
 1209 elements of this class.

1210 **Table 29 – Class: CIM_MemberOfCollection (Privilege)**

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

1211 **10.6 CIM_MemberOfCollection (Identity)**

1212 Table 30 contains the requirements for instances of CIM_MemberOfCollection when it is used to
 1213 associate instances of CIM_Identity with instances of CIM_Role.

1214 **Table 30 – Class: CIM_MemberOfCollection (Identity)**

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

1215 **10.7 CIM_OwningCollectionElement**

1216 CIM_OwningCollectionElement is used to associate an instance of CIM_Role with an instance of
 1217 CIM_ComputerSystem that represents the computer system to which the role belongs. Table 31 contains
 1218 the requirements for elements of this class.

1219 **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.

1220 **10.8 CIM_Privilege**

1221 CIM_Privilege is used to represent the privileges of a role. Table 32 contains the requirements for
 1222 elements of this class.

1223 **Table 32 – Class: CIM_Privilege**

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

1224 **10.9 CIM_RoleBasedManagementCapabilities**

1225 CIM_RoleBasedManagementCapabilities is used to indicate the capabilities for role-based privilege
 1226 management. Table 33 contains the requirements for elements of this class.

1227 **Table 33 – Class: CIM_RoleBasedManagementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
SharedPrivilegeSupported	Mandatory	See section 7.2.2.1.
ActivitiesSupported	Optional	See section 7.2.2.2.
ActivityQualifiersSupported	Optional	See section 7.2.2.2.
QualifierFormatsSupported	Optional	See section 7.2.2.2.
SupportedMethods	Mandatory	None
ElementName	Mandatory	Matches (pattern “. *”)

1228 **10.10 CIM_RegisteredProfile**

1229 The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. The requirements denoted
 1230 in Table 34 are in addition to those mandated by the *Profile Registration Profile*.

1231 **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”).

1232 **10.11 CIM_Role**

1233 CIM_Role is used to represent an authorized role. Table 35 contains the requirements for elements of this
 1234 class.

1235 **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 “.*”).

1236 **10.12 CIM_RoleBasedAuthorizationService**

1237 CIM_RoleBasedAuthorizationService is used to represent the service that handles the role management.
 1238 Table 36 contains the requirements for elements of this class.

1239 **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	See section 8.1.
DeleteRole()	Conditional	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.

1240 **10.13 CIM_RoleLimitedToTarget**

1241 CIM_RoleLimitedToTarget is used to associate an instance of CIM_Role with an instance of
 1242 CIM_ManagedElement that limits the scope of the role. Table 37 contains the requirements for elements
 1243 of this class.

1244 **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..*

1245 **10.14 CIM_ServiceAffectsElement**

1246 CIM_ServiceAffectsElement is used to associate an instance of CIM_RoleBasedAuthorizationService
 1247 with an instance of CIM_Role that represents the role that could be modified by using the service. Table
 1248 38 contains the requirements for elements of this class.

1249 **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
ElementAffects	Mandatory	Matches 5 (Manages)

1250 **10.15 CIM_ServiceServiceDependency**

1251 Table 39 contains the requirements for elements of this class.

1252 **Table 39 – 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

1253
1254
1255
1256
1257

Version	Date	Author	Description

1258 **ANNEX B**
1259 **(informative)**

1260
1261
1262 **Acknowledgements**

1263 The author wishes to acknowledge the following people.

1264 Editor:

- 1265 • Khachatur Papanyan – Dell
1266 • Aaron Merkin – IBM

1267 Contributors:

- 1268 • Jon Hass – Dell
1269 • Khachatur Papanyan – Dell
1270 • George Ericson – EMC
1271 • Christina Shaw – HP
1272 • Aaron Merkin – IBM