



1

2

3

4

Document Number: DSP1034

Date: 2009-06-17

Version: 1.0.1

5 **Simple Identity Management Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9

10 Copyright Notice

11 Copyright © 2008, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

33	Foreword	7
34	Introduction	8
35	1 Scope	9
36	2 Normative References.....	9
37	2.1 Approved References	9
38	2.2 Other References.....	9
39	3 Terms and Definitions	9
40	4 Symbols and Abbreviated Terms	11
41	5 Synopsis	11
42	6 Description	12
43	6.1 Authenticated Entities	13
44	6.2 Account	13
45	6.3 Account States.....	13
46	6.4 Local Account Security Policies.....	14
47	6.5 Access Ingress Point	14
48	6.6 Identity Context	14
49	7 Implementation.....	14
50	7.1 Base Requirements	14
51	7.2 Account Creation	17
52	7.3 Account Management.....	17
53	7.4 Representing a Third-Party Authenticated Principal.....	22
54	7.5 Managing Account Identity Groups.....	22
55	7.6 Representing Access Ingress Point.....	23
56	7.7 Identity Context	23
57	8 Methods.....	23
58	8.1 CIM_AccountManagementService.CreateAccount()	23
59	8.2 CIM_Account.RequestStateChange()	25
60	8.3 Profile Conventions for Operations.....	27
61	8.4 CIM_Account	27
62	8.5 CIM_EnabledLogicalElementCapabilities.....	28
63	8.6 CIM_AccountOnSystem.....	28
64	8.7 CIM_AccountManagementCapabilities.....	29
65	8.8 CIM_AccountManagementService	29
66	8.9 CIM_AccountSettingData	29
67	8.10 CIM_AssignedIdentity	30
68	8.11 CIM_Dependency	30
69	8.12 CIM_ElementCapabilities	30
70	8.13 CIM_ElementSettingData	31
71	8.14 CIM_Group	31
72	8.15 CIM_HostedService	31
73	8.16 CIM_Identity	32
74	8.17 CIM_IdentityContext	32
75	8.18 CIM_MemberOfCollection	32
76	8.19 CIM_OwningCollectionElement.....	33
77	8.20 CIM_ServiceAffectsElement	33
78	8.21 CIM_SettingsDefineCapabilities	33
79	8.22 CIM_UserContact	34
80	9 Use Cases.....	34
81	9.1 Profile Registration.....	34
82	9.2 Determine Whether CIM_Account.ElementName Can Be Modified	43
83	9.3 Determine Whether Account State Management Is Supported	43
84	9.4 Determine Whether Account Management Is Supported.....	43

85	9.5	Create an Account	43
86	9.6	Determine Account Defaults	44
87	9.7	Delete an Account.....	44
88	9.8	Modify the Password for an Account	44
89	9.9	Clear an Account	45
90	9.10	Change State to Enabled Offline	45
91	9.11	Add an Account Identity to a Group.....	45
92	9.12	Remove an Account Identity from a Group	45
93	9.13	Determine the Context of a Security Principal.....	45
94	10	CIM Elements.....	46
95	10.1	CIM_Account	47
96	10.2	CIM_AccountManagementCapabilities.....	47
97	10.3	CIM_AccountManagementService	48
98	10.4	CIM_AccountOnSystem.....	48
99	10.5	CIM_AccountSettingData	48
100	10.6	CIM_AssignedIdentity (CIM_Account).....	49
101	10.7	CIM_AssignedIdentity (Group)	49
102	10.8	CIM_AssignedIdentity (UserContact)	49
103	10.9	CIM_Dependency (Access Ingress)	49
104	10.10	CIM_ElementCapabilities (CIM_AccountManagementService)	50
105	10.11	CIM_ElementCapabilities (CIM_Account)	50
106	10.12	CIM_ElementSettingData	50
107	10.13	CIM_EnabledLogicalElementCapabilities.....	51
108	10.14	CIM_Group	51
109	10.15	CIM_HostedService	51
110	10.16	CIM_Identity.....	51
111	10.17	CIM_IdentityContext	52
112	10.18	CIM_MemberOfCollection (Group Membership)	52
113	10.19	CIM_OwningCollectionElement.....	52
114	10.20	CIM_ServiceAffectsElement	53
115	10.21	CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)	53
116	10.22	CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)	53
117	10.23	CIM_UserContact	54
118	10.24	CIM_RegisteredProfile.....	54
119	ANNEX A (informative)	Change Log.....	55
120			

121 Figures

122	Figure 1 – <i>Simple Identity Management Profile</i> : Class Diagram	12
123	Figure 2 – Profile Registration.....	34
124	Figure 3 – Basic System Accounts	35
125	Figure 4 – Full Account Capabilities	36
126	Figure 5 – Account Capabilities with Ranges	37
127	Figure 6 – Third-Party Authenticated User	38
128	Figure 7 – Accounts with Group Membership.....	39
129	Figure 8 – Role-Oriented Groups.....	41
130	Figure 9 – Access Ingress Point and Identity Context.....	42
131		

132 **Tables**

133 Table 1 – Referenced Profiles 12

134 Table 2 – CIM_AccountManagementService.CreateAccount() Method: Return Code Values 24

135 Table 3 – CIM_AccountManagementService.CreateAccount() Method: Parameters 24

136 Table 4 – CIM_Account.RequestStateChange() Method: Return Code Values 26

137 Table 5 – CIM_Account.RequestStateChange() Method: Parameters 26

138 Table 6 – Operations: CIM_Account 27

139 Table 7 – Operations: CIM_AccountOnSystem 28

140 Table 8 – Operations: CIM_AccountManagementService 29

141 Table 9 – Operations: CIM_AccountSettingData 29

142 Table 10 – Operations: CIM_AssignedIdentity 30

143 Table 11 – Operations: CIM_Dependency 30

144 Table 12 – Operations: CIM_ElementCapabilities 30

145 Table 13 – Operations: CIM_ElementSettingData 31

146 Table 14 – Operations: CIM_HostedService 32

147 Table 15 – Operations: CIM_IdentityContext 32

148 Table 16 – Operations: CIM_MemberOfCollection 32

149 Table 17 – Operations: CIM_OwningCollectionElement 33

150 Table 18 – Operations: CIM_ServiceAffectsElement 33

151 Table 19 – Operations: CIM_SettingsDefineCapabilities 34

152 Table 20 – CIM Elements: *Simple Identity Management Profile* 46

153 Table 21 – Class: CIM_Account 47

154 Table 22 – Class: CIM_AccountManagementCapabilities 47

155 Table 23 – Class: CIM_AccountManagementService 48

156 Table 24 – Class: CIM_AccountOnSystem 48

157 Table 25 – Class: CIM_AccountSettingData 48

158 Table 26 – Class: CIM_AssignedIdentity (CIM_Account) 49

159 Table 27 – Class: CIM_AssignedIdentity (Group) 49

160 Table 28 – Class: CIM_AssignedIdentity (UserContact) 49

161 Table 29 – Class: CIM_Dependency (Access Ingress) 49

162 Table 30 – Class: CIM_ElementCapabilities (CIM_AccountManagementService) 50

163 Table 31 – Class: CIM_ElementCapabilities (CIM_Account) 50

164 Table 32 – Class: CIM_ElementSettingData 50

165 Table 33 – Class: CIM_EnabledLogicalElementCapabilities 51

166 Table 34 – Class: CIM_Group 51

167 Table 35 – Class: CIM_HostedService 51

168 Table 36 – Class: CIM_Identity 51

169 Table 37 – Class: CIM_IdentityContext 52

170 Table 38 – Class: CIM_MemberOfCollection (Group Membership) 52

171 Table 39 – Class: CIM_OwningCollectionElement 52

172 Table 40 – Class: CIM_ServiceAffectsElement (Account) 53

173 Table 41 – Class: CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities) 53

174 Table 42 – Class: CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities) 54

175 Table 43 – Class: CIM_UserContact 54

176 Table 44 – Class: CIM_RegisteredProfile 54

177

179

Foreword

180 The *Simple Identity Management Profile* (DSP1034) was prepared by the Security Working Group, the
181 Physical Platform Profiles Working Group, the Server Management Working Group, and the WBEM
182 Infrastructure Modeling Working Group of the DMTF.

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

185

186 **Acknowledgments**

187 The authors wish to acknowledge the following people.

188 Authors:

- 189 • Aaron Merkin – IBM
- 190 • Murali Rajagopal – Broadcom
- 191 • Hemal Shah – Broadcom

192 Contributors:

- 193 • Jon Hass – Dell
- 194 • Khachatur Papanyan – Dell
- 195 • George Ericson – EMC
- 196 • Christina Shaw – HP
- 197 • David Hines – Intel

198

Introduction

199 The information in this specification should be sufficient for a provider or consumer of this data to identify
200 unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to
201 represent and manage an Account and its Security Principal that is modeled using the DMTF Common
202 Information Model (CIM) core and extended model definitions.

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

205

Simple Identity Management Profile

206 1 Scope

207 The *Simple Identity Management Profile* is a component profile that provides the ability to manage local
208 accounts on a system and to represent the local system's view of a principal that is authenticated through
209 a third-party authentication service. This profile does not specify CIM-based mechanisms for performing
210 the authentication of credentials.

211 2 Normative References

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

215 2.1 Approved References

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

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

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

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

224 DMTF DSP1039, *Role Based Authorization Profile 1.0*,
225 http://www.dmtf.org/standards/published_documents/DSP1039_1.0.pdf

226 ANSI T1.276-2003, *Operations, Administration, Maintenance, and Provisioning Security Requirements for*
227 *the Public Telecommunications Network: A Baseline of Security Requirements for the Management*
228 *Plane*, <http://webstore.ansi.org>

229 2.2 Other References

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

232 3 Terms and Definitions

233 For the purposes of this document, the following terms and definitions apply. For the purposes of this
234 document, the terms and definitions given in [DSP1033](#), [DSP1001](#), and [T1.276-2003](#) also apply.

235 3.1

236 **account identity**

237 the security principal that represents an authenticated Account.

- 238 **3.2**
239 **can**
240 used for statements of possibility and capability, whether material, physical, or causal
- 241 **3.3**
242 **cannot**
243 used for statements of possibility and capability, whether material, physical, or causal
- 244 **3.4**
245 **conditional**
246 indicates requirements to be followed strictly in order to conform to the document when the specified
247 conditions are met
- 248 **3.5**
249 **mandatory**
250 indicates requirements to be followed strictly in order to conform to the document and from which no
251 deviation is permitted
- 252 **3.6**
253 **may**
254 indicates a course of action permissible within the limits of the document
- 255 **3.7**
256 **need not**
257 indicates a course of action permissible within the limits of the document
- 258 **3.8**
259 **optional**
260 indicates a course of action permissible within the limits of the document
- 261 **3.9**
262 **referencing profile**
263 indicates a profile that owns the definition of this class and can include a reference to this profile in its
264 "Referenced Profiles" table
- 265 **3.10**
266 **shall**
267 indicates requirements to be followed strictly in order to conform to the document and from which no
268 deviation is permitted
- 269 **3.11**
270 **shall not**
271 indicates requirements to be followed in order to conform to the document and from which no deviation is
272 permitted
- 273 **3.12**
274 **should**
275 indicates that among several possibilities, one is recommended as particularly suitable, without
276 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 277 **3.13**
278 **should not**
279 indicates that a certain possibility or course of action is deprecated but not prohibited

280 **3.14**
281 **unspecified**
282 indicates that this profile does not define any constraints for the referenced CIM element or operation

283 **3.15**
284 **authentication**
285 the process of verifying the credentials provided by an entity for the purpose of resolving to a security
286 principal

287 **3.16**
288 **first-party authentication**
289 authentication that is performed using services that execute local to the relying party

290 **3.17**
291 **principal**
292 an entity that can be positively identified and verified through an authentication mechanism

293 **3.18**
294 **third-party authentication**
295 authentication that is performed using services that execute remote to the relying party

296 **4 Symbols and Abbreviated Terms**

297 The following abbreviations are used in this document.

298 **4.1**
299 **CIM**
300 Common Information Model

301 **5 Synopsis**

302 **Profile Name:** *Simple Identity Management*

303 **Version:** 1.0.1

304 **Organization:** DMTF

305 **CIM schema version:** 2.22

306 **Central Class:** CIM_AccountManagementService

307 **Scoping Class:** CIM_ComputerSystem

308 The *Simple Identity Management Profile* extends the management capability of the referencing profiles by
309 adding the capability to describe management of user accounts.

310 CIM_AccountManagementService shall be the Central Class of this profile. The instance of
311 CIM_AccountManagementService shall be the Central Instance of this profile. CIM_ComputerSystem
312 shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem with which the Central
313 Instance is associated through an instance of CIM_HostedService shall be the Scoping Instance of this
314 profile.

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

316

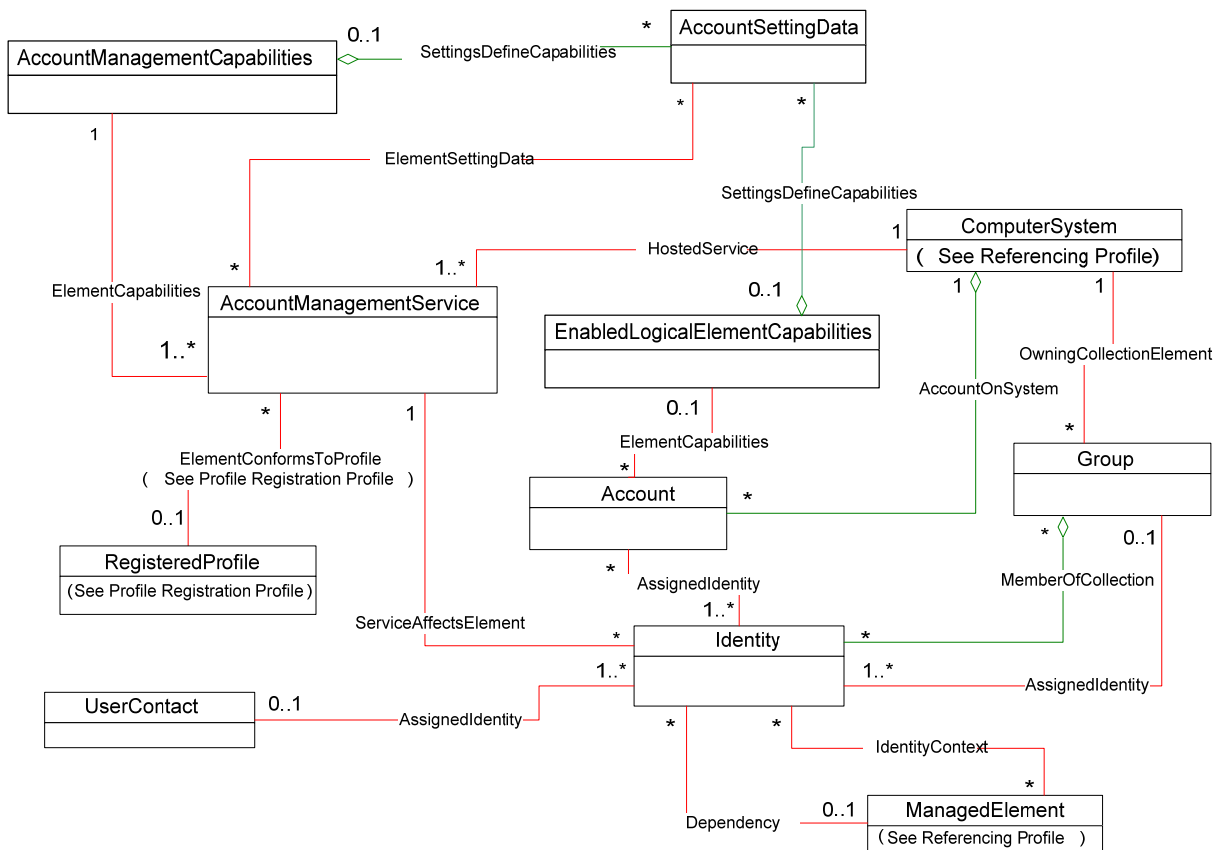
Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

317 **6 Description**

318 The *Simple Identity Management Profile* provides the ability to perform management of user accounts of
 319 a system that use basic user ID and password authentication. This profile also provides the ability to
 320 represent a principal with an UserID and that has been authenticated through third-party authentication.

321 Figure 1 represents the class schema for the *Simple Identity Management Profile*. For simplicity, the
 322 prefix *CIM_* has been removed from the names of the classes.



323

324 **Figure 1 – Simple Identity Management Profile: Class Diagram**

325 The *CIM_AccountManagementService* provides the ability to manage accounts on the system.
 326 *CIM_Account* represents accounts that are defined locally on the system. *CIM_Group* provides the ability
 327 to group account identities for authorization purposes. *CIM_UserContact* provides descriptive information
 328 about an individual who has been authenticated through third-party authentication. *CIM_Identity*
 329 represents a security principal. The *CIM_AssignedIdentity* association is used to associate the security
 330 principal with the entity whose privileges are being managed. Local accounts, third-party authenticated
 331 users, and account identity groups each can have one or more associated security principals. These

332 security principles create a relationship between the authenticated individual and the authorization
333 granted to the individual.

334 NOTE: CIM_Group may provide the ability to group other identities in future but this specification only supports
335 grouping account identities.

336 **6.1 Authenticated Entities**

337 This profile identifies requirements for modeling three types of authenticated entities: local accounts,
338 third-party authenticated entities, and account groups. Local accounts are modeled using CIM_Account.
339 Third-party authenticated users may be modeled with instances of CIM_UserContact. Together with
340 CIM_AssignedIdentity this provides an explicit means to model who an Identity represents. Identity
341 groups are modeled with CIM_Group.

342 This profile provides support for adding and removing local accounts. Therefore, when account
343 management is supported, it is possible to be in an intermediate state in which no local accounts are
344 defined.

345 A common implementation of authentication and authorization support is for a local system to use a
346 security client to perform the authentication of credentials in conjunction with a third-party authentication
347 service. Some implementations perform their privilege management using a third-party service as well.
348 These two services can be combined such that the local system passes credentials to a third-party
349 service and upon successful validation receives information about the privileges associated with those
350 credentials in return. The local system persists no information about the authenticated entity, and the
351 knowledge of the entity and its privileges are transient with existence of the underlying secure session
352 established with the system. The support for modeling third-party authenticated users provides the ability
353 to represent the system's transient knowledge. An effect of modeling this transient knowledge is that even
354 when the optional behavior of modeling third-party authenticated users is supported, zero instances of
355 CIM_UserContact can exist at any point in time.

356 This profile does not provide support for adding or removing account identity groups. Therefore, when
357 group management is supported, at least one instance of CIM_Group exists.

358 **6.2 Account**

359 Instances of the CIM_Account class provide an interface to locally stored authentication information, such
360 as used by a Unix or Windows login. The interface does not provide accounting information such as: a
361 history of when a user was logged into a system; or billing information.

362 **6.3 Account States**

363 Accounts on a system have four common states: enabled, disabled, offline, and quiesce.

364 When an account is enabled, it is properly configured and available for use. The authentication service
365 will attempt to validate credentials against it.

366 When the account is in a disabled state, it is unavailable for authentication. The account may or may not
367 be properly configured.

368 NOTE 1: Some systems maintain a fixed number of accounts. Rather than add and remove the account from the
369 system when it is not in use, it is placed in the disabled state. When the account is in this state, it is effectively
370 unavailable for authorization against it. The account can be configured and then enabled.

371 When an account is in offline state, it is properly configured and conforms to currently implemented
372 security policies but is unavailable for authentication.

373 NOTE 2: Some accounts may enter the offline state from the disabled state before entering the enabled state. Some
374 accounts may enter the offline state from the enabled state for administrative reasons.

375 When an account is in the quiesce state (locked-out) it is properly configured but may not conform to
376 currently implemented security policies and it is not available for authentication.

377 NOTE 3: This state is usually the result of a violation of a system policy. Before access can be granted to the
378 resources secured by the account, corrective action is required in this case. For example, an account can be placed
379 into the locked-out state because the password expired, the number of consecutive failed access attempts exceeded
380 the limit set by policy, the inactivity period exceeded the limit set by policy, and so on. This action can be taken by the
381 user to whom the account corresponds (for example, a changing the password), or it can be an administrative action.

382 The account state is modeled using the EnabledState property of CIM_Account.

383 **6.4 Local Account Security Policies**

384 Systems often have account policies in place to enhance the security associated with local account
385 authorization. Examples of these policies include password complexity requirements, password expiration
386 limits, limits on consecutive failed access attempts, and so on. These policies generally have
387 configuration parameters associated with them. For example, if a system supports a policy of enforcing a
388 password expiration date, the policy could require the password to change every 90 days.

389 CIM_EnabledLogicalElementCapabilities is used with CIM_AccountSettingData to indicate additional
390 account policies supported for a specific account. The parameters for the policy are provided by
391 properties of the CIM_Account instance. CIM_AccountSettingData used in conjunction with
392 CIM_AccountManagementCapabilities indicates the policies and their parameters that are enforced when
393 creating an account. CIM_AccountSettingData is also used to indicate default values for properties of a
394 CIM_Account instance if they are not provided by the client when the CIM_Account is created.

395 **6.5 Access Ingress Point**

396 Access to a system can be provided over one or more interfaces. When access for a security principal is
397 authenticated over an interface, the interface can be identified.

398 When CIM_Dependency references an instance of CIM_Identity and an instance of a subclass of
399 CIM_ManagedElement other than CIM_Role, it is used to indicate that the security principal represented
400 by the CIM_Identity instance is authenticated over or through the referenced CIM_ManagedElement.

401 **6.6 Identity Context**

402 An account, account identity group, or third-party authenticated entity can have more than one security
403 principal associated with it. The security principals are frequently differentiated based on the mechanism
404 through which the credentials that identify the underlying entity were supplied. For example, credentials
405 validated against an account on a system could resolve to a different security principal depending on
406 whether the credentials were supplied over a terminal session, through a remote management interface,
407 or locally. The security principals can have different privileges assigned to them. The need to manage
408 privileges for an authenticated entity that vary based on context is a common reason for having multiple
409 security principals associated with the authenticated entity.

410 **7 Implementation**

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

413 **7.1 Base Requirements**

414 This section describes the requirements that are common for all implementations of the profile.

415 Zero or more instances of CIM_Identity representing security principals shall exist (see sections 7.1.3,
416 7.4.1, and 7.5.1).

417 **7.1.1 CIM_AccountManagementService**

418 At least one instance of CIM_AccountManagementService shall exist.

419 **7.1.1.1 CIM_AccountManagementService.ElementName Constraints**

420 The ElementName property of CIM_AccountManagementService may be modifiable by a client or it may
421 have a fixed value.

422 **7.1.1.1.1 ElementName Is Not Modifiable**

423 The ElementNameEditSupported property shall have a value of FALSE when the implementation does
424 not support client modification of the CIM_AccountManagementService.ElementName property. When an
425 implementation does not support modification of the ElementName property by a client, the
426 ElementName property shall be formatted as a free-form string of variable length (pattern ".*").

427 **7.1.1.1.2 ElementName Is Modifiable**

428 The CIM_AccountManagementService.ElementName property may be modified by a client. This behavior
429 is conditional. This section describes the CIM elements and behavioral requirements when an
430 implementation supports client modification of the CIM_AccountManagementService.ElementName
431 property.

432 **7.1.2 CIM_AccountManagementCapabilities**

433 Exactly one instance of CIM_AccountManagementCapabilities shall be associated with each instance of
434 CIM_AccountManagementService through the CIM_ElementCapabilities association.

435 **7.1.2.1 CIM_AccountManagementCapabilities.ElementNameEditSupported**

436 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
437 client modification of the CIM_AccountManagementService.ElementName property.

438 **7.1.2.2 CIM_AccountManagementCapabilities.MaxElementNameLen**

439 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
440 property has a value of TRUE. The MaxElementNameLen property shall indicate the maximum length of
441 a string that the implementation will accept as a value for the ElementName property of the associated
442 CIM_AccountManagementService instance.

443 **7.1.3 CIM_Account**

444 CIM_Account shall represent an account on a managed system, where CIM_ComputerSystem represents
445 the managed system and is associated to CIM_Account through the CIM_AccountOnSystem association.
446 CIM_Account shall be associated to CIM_Identity that represents the account's security principal through
447 CIM_AssignedIdentity association. CIM_Account is scoped to the Central Instance through this
448 CIM_Identity, which is associated to the Central Instance through the CIM_ServiceAffectsElement
449 association.

450 If CIM_AccountManagementCapabilities.OperationsSupported contains one of these values: 2 (Create), 3
451 (Modify), or 4 (Delete), then CIM_Account, CIM_AccountOnSystem and CIM_AssignedIdentity shall be
452 supported.

453 **7.1.3.1 CIM_Account.UserPassword Constraints**

454 The UserPassword property of CIM_Account may be clear text or it may be encrypted.

455 When an instance of CIM_Account is retrieved and the underlying account has a valid password, the
456 value of the CIM_Account.UserPassword property shall be an array of length zero to indicate that the
457 account has a password configured.

458 When the underlying account does not have a valid password, the CIM_Account.UserPassword property
459 shall be NULL.

460 The following two sections describe the requirements for setting the CIM_Account.UserPassword.

461 **7.1.3.1.1 UserPassword Is Clear Text**

462 When the SupportedUserPasswordEncryptionAlgorithms[] property of
463 CIM_AccountManagementCapabilities is NULL, UserPassword shall be clear text and
464 UserPasswordEncryptionAlgorithm shall have no value.

465 When the SupportedUserPasswordEncryptionAlgorithms[] property of
466 CIM_AccountManagementCapabilities has no values, UserPassword shall be clear text and
467 UserPasswordEncryptionAlgorithm shall have no value.

468 When the SupportedUserPasswordEncryptionAlgorithms[] property of
469 CIM_AccountManagementCapabilities only has the value 0 (None), UserPassword shall be clear text and
470 UserPasswordEncryptionAlgorithm shall have the value 0 (None).

471 When the SupportedUserPasswordEncryptionAlgorithms[] property of
472 CIM_AccountManagementCapabilities has several values, including the value 0 (None), UserPassword
473 may be clear text. In this case when UserPassword is in clear text, UserPasswordEncryptionAlgorithm
474 shall have the value 0 (None).

475 **7.1.3.1.2 UserPassword Is Encrypted**

476 When the SupportedUserPasswordEncryptionAlgorithms[] property of
477 CIM_AccountManagementCapabilities contains one or more values but not 0 (None), UserPassword shall
478 be encrypted.

479 When the SupportedUserPasswordEncryptionAlgorithms[] property of
480 CIM_AccountManagementCapabilities contains zero and non-zero values, UserPassword may be
481 encrypted.

482 When UserPassword is encrypted, it shall be encrypted in one of the forms specified by the value of the
483 SupportedUserPasswordEncryptionAlgorithms[] property and UserPasswordEncryptionAlgorithm shall
484 have a value corresponding to that form of encryption.

485 **7.1.3.2 UserID/UserPassword Usage for Authentication**

486 An instance of CIM_Account can be used for user ID/password based authentication. If an instance of
487 CIM_Account is used for user ID/password based authentication, the following rules apply:

- 488 1) The value of CIM_Account.UserID shall be used as the user ID for the authentication.
- 489 2) The currently set value of CIM_Account.UserPassword shall be used as the password for the
490 authentication.

491 **7.1.4 Representing a Security Principal**

492 Each security principal shall be represented with an instance of CIM_Identity. Each instance of
493 CIM_Identity shall be associated with exactly one instance of CIM_AccountManagementService through
494 the CIM_ServiceAffectsElement association.

495 **7.1.5 At Least One Authentication Model**

496 At least one of the optional behaviors specified by sections 7.3, 7.4, and 7.5 shall be supported.

497 **7.2 Account Creation**

498 The ability to create accounts by using the `CIM_AccountManagementService.CreateAccount()` method
499 may be supported. This behavior is conditional. See section 8.1 for a description of the method.

500 This section details additional requirements that are conditional on support for account creation. These
501 requirements shall be supported when the `CIM_AccountManagementCapabilities.OperationsSupported`
502 property of the instance of `CIM_AccountManagementCapabilities` that is associated with the
503 `CIM_AccountManagementService` through the `CIM_ElementCapabilities` association contains the value 2
504 (Create).

505 **7.2.1 Modeling Account Defaults**

506 The default property values for an instance of `CIM_Account` that is created by invoking the
507 `CIM_AccountManagementService.CreateAccount()` method may be modeled. This behavior is optional.
508 When this behavior is implemented, the requirements specified in this section shall be met.

509 Zero or more instances of `CIM_AccountSettingData` may be associated with an instance of
510 `CIM_AccountManagementService` through the `CIM_ElementSettingData` association. These instances of
511 `CIM_AccountSettingData` are used to provide default values for instances of `CIM_Account` that are
512 created by `CIM_AccountManagementService.CreateAccount()` method.

513 At most one instance of `CIM_AccountSettingData` shall be associated with an instance of
514 `CIM_AccountManagementService` through an instance of `CIM_ElementSettingData` where the
515 `CIM_ElementSettingData.IsNext` property has the value 1 (Is Next). This instance of
516 `CIM_AccountSettingData` contains the default values for properties of a created instance of `CIM_Account`.
517 Section 8.1 describes the use of this instance when the
518 `CIM_AccountManagementService.CreateAccount()` method is invoked. Other instances of
519 `CIM_AccountSettingData` may be associated with `CIM_AccountManagementService` through an instance
520 of `CIM_ElementSettingData` and shall have the `CIM_ElementSettingData.IsNext` property not set to 1 (Is
521 Next).

522 **7.2.2 Capabilities and Requirements for Account Creation**

523 Requirements and capabilities for instances of `CIM_Account` that are created by using the
524 `CIM_AccountManagementService.CreateAccount()` method may be modeled according to the
525 requirements specified in section 7.3.5 where the instance of `CIM_Capabilities` is the instance of
526 `CIM_AccountManagementCapabilities` that is associated with the `CIM_AccountManagementService`
527 instance.

528 **7.3 Account Management**

529 Support for managing accounts on a system is optional behavior. This section details the requirements
530 that shall be met when this behavior is implemented.

531 Zero or more instances of `CIM_Account` shall be associated with the Scoping Instance through the
532 `CIM_AccountOnSystem` association.

533 **7.3.1 Identity for an Account**

534 One or more instances of `CIM_Identity` shall be associated with an instance of `CIM_Account` through the
535 `CIM_AssignedIdentity` association.

536 **7.3.2 Capabilities of an Account**

537 Zero or one instances of CIM_EnabledLogicalElementCapabilities shall be associated with an instance of
538 CIM_Account through the CIM_ElementCapabilities association.

539 Additional capabilities of an instance of CIM_Account may be modeled using the requirements specified
540 in section 7.3.5 where the instance of CIM_Capabilities is an instance of
541 CIM_EnabledLogicalElementCapabilities associated with the instance of CIM_Account.

542 If an instance of CIM_EnabledLogicalElementCapabilities representing the capabilities of an account is
543 instantiated, then that instance shall be associated via CIM_ElementCapabilities with the instance of
544 CIM_Account that represents that account.

545 **7.3.3 Managing the Account's State**

546 This section describes the use of the RequestedState and EnabledState properties to represent the state
547 of an instance of CIM_Account.

548 **7.3.3.1 State Management Supported**

549 Support for managing the state of the CIM_Account instance is conditional behavior. This section
550 describes the CIM elements and behaviors that shall be implemented when this behavior is supported.

551 **7.3.3.2 CIM_Account.RequestStateChange() Supported**

552 When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least
553 one value, the CIM_Account.RequestStateChange() method shall be implemented and supported. The
554 CIM_Account.RequestStateChange() method shall not return a value of 1 (Not Supported).

555 **7.3.3.3 CIM_Account.RequestedState**

556 If the CIM_Account.RequestStateChange() method is successfully invoked, the value of the
557 RequestedState property shall be the value of the RequestedState parameter. If the method is not
558 successfully invoked, the value of the RequestedState property is indeterminate. When the
559 RequestedStatesSupported property of the associated instance of
560 CIM_EnabledLogicalElementCapabilities contains one or more values, the RequestedState property shall
561 have one of the values specified or a value of 5 (No Change). When the RequestedStatesSupported
562 property of the associated instance of CIM_EnabledLogicalElementCapabilities does not contain any
563 values, the RequestedState property shall have the value of 12 (Not Applicable).

564 **7.3.3.4 CIM_Account.EnabledState**

565 The Account State is modeled using the EnabledState property of CIM_Account (see 6.3).

566 When the RequestedState parameter has a value of 2 (Enabled), 3 (Disabled), or 6 (Offline) after
567 successful completion of the CIM_Account.RequestStateChange() method, the value of the
568 EnabledState property shall equal the value of the RequestedState property. If the method does not
569 complete successfully, the value of the EnabledState property is indeterminate. The EnabledState
570 property shall have the value 2 (Enabled), 3 (Disabled), 6 (Enabled but Offline), or 5 (Not Applicable).

571 A value of 2 (Enabled) shall indicate that the account is properly configured and is enabled for use. An
572 attempt to authenticate against the credentials of the account will be processed.

573 A value of 3 (Disabled) shall indicate that the account is disabled for use and attempts to authenticate
574 against the credentials of the account will not be processed. After the account has transitioned to
575 3 (Disabled), the account may not be properly configured. The account may be properly configured but is
576 not required to be. Thus a transition to 2 (Enabled) may not succeed without a reconfiguration of the
577 account.

578 A value of 6 (Enabled but Offline) shall indicate that the account is properly configured but is not enabled
579 for use. An attempt to authenticate against the credentials of the account will not be processed. A
580 transition back to 2 (Enabled) should succeed without requiring configuration of the account.

581 A value of 9 (Quiesce) shall indicate that the account is in a locked-out state and requires corrective
582 action to restore it to operational usage. The corrective action required and the mechanism through which
583 it is undertaken is undefined. Note that this state is not entered as a result of RequestStateChange()
584 method transition.

585 When disabling of an account is supported without the ability to further distinguish between disablement
586 with the clearing of the account configuration and disablement without the clearing of the account
587 configuration, the value 3 (Disabled) shall be used and the value 6 (Enabled but Offline) shall not be
588 used.

589 **7.3.3.5 Indicating State Management Support with CIM_EnabledLogicalElementCapabilities**

590 When state management is supported, the RequestedStatesSupported property of the
591 CIM_EnabledLogicalElementCapabilities instance associated with the CIM_Account instance through an
592 instance of CIM_ElementCapabilities shall contain at least one value. The RequestedStatesSupported
593 property may have zero or more of the following values: 2 (Enabled), 3 (Disabled), or 6 (Offline).

594 **7.3.4 CIM_Account.ElementName Constraints**

595 The ElementName property of CIM_Account may be modifiable by a client or it may have a fixed value.

596 **7.3.4.1 ElementName Is Not Modifiable**

597 The ElementNameEditSupported property shall have a value of FALSE when the implementation does
598 not support client modification of the CIM_Account.ElementName property.

599 When an implementation does not support modification of the ElementName property by a client, the
600 ElementName property shall be formatted as a free-form string of variable length (pattern ".*").

601 **7.3.4.2 ElementName Is Modifiable**

602 The CIM_Account.ElementName property may be modified by a client. This behavior is conditional. This
603 section describes the CIM elements and behavioral requirements when an implementation supports client
604 modification of the CIM_Account.ElementName property.

605 **7.3.4.2.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported**

606 The ElementNameEditSupported property shall have a value of TRUE when the implementation supports
607 client modification of the CIM_Account.ElementName property.

608 **7.3.4.2.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen**

609 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
610 property has a value of TRUE. The MaxElementNameLen property shall indicate the maximum length of
611 a string that the implementation will accept as a value for the ElementName property of the associated
612 CIM_Account instance.

613 **7.3.4.2.3 CIM_EnabledLogicalElementCapabilities.ElementNameMask**

614 The ElementNameMask property shall be implemented when the ElementNameEditSupported property
615 has a value of TRUE. The ElementNameMask property shall contain a regular expression defined using
616 the syntax specified in Annex C of [DSP1001](#).

617 **7.3.5 Modeling Account Requirements and Capabilities**

618 Constraints on the property values of an instance of CIM_Account may be modeled. This behavior is
619 optional. The requirements specified in this section shall be met when this behavior is implemented.

620 This section describes how constraints for properties of an instance of CIM_Account may be modeled
621 using instances of CIM_AccountSettingData that are associated with an instance of
622 CIM_EnabledLogicalElementCapabilities through an instance of CIM_SettingsDefineCapabilities. One or
623 more instances of CIM_AccountSettingData may be associated with an instance of
624 CIM_EnabledLogicalElementCapabilities through the CIM_SettingsDefineCapabilities association.

625 **7.3.5.1 Password History Depth**

626 The following requirements shall be met when the PasswordHistoryDepth property of an instance of
627 CIM_AccountSettingData that is associated with the CIM_EnabledLogicalElementCapabilities instance
628 through the CIM_SettingsDefineCapabilities association has a non-Null value.

629 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the value
630 of the PasswordHistoryDepth property shall represent the maximum value that is supported for the
631 CIM_Account.PasswordHistoryDepth property.

632 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the value of
633 the PasswordHistoryDepth property shall represent the minimum value that is supported for the
634 CIM_Account.PasswordHistoryDepth property.

635 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 0 (Point), the value of the
636 PasswordHistoryDepth property shall represent the only value that is supported for the
637 CIM_Account.PasswordHistoryDepth property.

638 **7.3.5.2 Password Expiration**

639 The following requirements shall be met when the MaximumPasswordExpiration property of an instance
640 of CIM_AccountSettingData that is associated with the CIM_EnabledLogicalElementCapabilities instance
641 through the CIM_SettingsDefineCapabilities association has a non-Null value.

642 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the value
643 of the MaximumPasswordExpiration property shall represent the maximum value expressed as an interval
644 that is supported for the CIM_Account.PasswordExpiration property.

645 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the date-
646 time value that results from adding the value of the MaximumPasswordExpiration property to the current
647 date-time shall represent the maximum date-time value that is supported for the
648 CIM_Account.PasswordExpiration property.

649 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the value of
650 the MaximumPasswordExpiration property shall represent the minimum value expressed as an interval
651 that is supported for the CIM_Account.PasswordExpiration property.

652 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the date-
653 time value that results from adding the value of the MaximumPasswordExpiration property to the current
654 date-time shall represent the minimum date-time value that is supported for the
655 CIM_Account.PasswordExpiration property.

656 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 0 (Point), the value of the
657 MaximumPasswordExpiration property shall represent the only value that is supported for the
658 CIM_Account.PasswordExpiration property.

659 7.3.5.3 Complex Password Rules

660 The following requirements shall be met when the ComplexPasswordRulesEnforced property of an
661 instance of CIM_AccountSettingData that is associated with the CIM_Capabilities instance through the
662 CIM_SettingsDefineCapabilities association has a non-Null value.

663 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the values
664 contained in the ComplexPasswordRulesEnforced property shall represent the minimum set of values
665 that are required to be contained in the CIM_Account.ComplexPasswordRulesEnforced property for the
666 instance of CIM_AccountManagementService that is associated with the CIM_Capabilities instance.

667 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 0 (Points), the value of the
668 ComplexPasswordRulesEnforced property shall represent the only combination of values supported for
669 the CIM_Account.ComplexPasswordRulesEnforced property for the instance of
670 CIM_AccountManagementService that is associated with the CIM_Capabilities instance.

671 7.3.5.4 Inactivity Timeout

672 The following requirements shall be met when the InactivityTimeout property of an instance of
673 CIM_AccountSettingData that is associated with the CIM_Capabilities instance through the
674 CIM_SettingsDefineCapabilities association has a non-Null value.

675 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the value
676 of the InactivityTimeout property shall represent the maximum value expressed as an interval that is
677 supported for the CIM_Account.InactivityTimeout property.

678 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the date-
679 time value that results from adding the value of the InactivityTimeout property to the current date-time
680 shall represent the maximum date-time value that is supported for the CIM_Account.InactivityTimeout
681 property.

682 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the value of
683 the InactivityTimeout property shall represent the minimum value expressed as an interval that is
684 supported for the CIM_Account.InactivityTimeout property.

685 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the date-
686 time value that results from adding the value of the InactivityTimeout property to the current date-time
687 shall represent the minimum date-time value that is supported for the CIM_Account.InactivityTimeout
688 property.

689 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 0 (Point), the value of the
690 InactivityTimeout property shall represent the only value that is supported for the
691 CIM_Account.InactivityTimeout property.

692 Note: Account State (see 6.2) may change due to inactivity timeout expiry set by this property.

693 7.3.5.5 Successive Failed Logins

694 The following requirements shall be met when the MaximumSuccessiveLoginFailures property of an
695 instance of CIM_AccountSettingData that is associated with the CIM_Capabilities instance through the
696 CIM_SettingsDefineCapabilities association has a non-Null value.

697 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the value
698 of the MaximumSuccessiveLoginFailures property shall represent the maximum value that is supported
699 for the CIM_Account.MaximumSuccessiveLoginFailures property.

700 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the value of
701 the MaximumSuccessiveLoginFailures property shall represent the minimum value that is supported for
702 the CIM_Account.MaximumSuccessiveLoginFailures property.

703 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 0 (Point), the value of the
704 MaximumSuccessiveLoginFailures property shall represent the only value that is supported for the
705 CIM_Account.MaximumSuccessiveLoginFailures property.

706 Note: Account State (see 6.2) may change after the consecutive failed login attempts set by this property.

707 **7.4 Representing a Third-Party Authenticated Principal**

708 User information about an identity that has been authenticated through a third-party authentication
709 service may be modeled. This behavior is optional. This section describes the requirements when this
710 user information is modeled. This user information shall be modeled using an instance of
711 CIM_UserContact. Zero or more instances of CIM_UserContact shall exist.

712 **7.4.1 Identity for CIM_UserContact**

713 One or more instances of CIM_Identity shall be associated with an instance of CIM_UserContact through
714 the CIM_AssignedIdentity association.

715 **7.4.2 Profile Conformance Scope for CIM_UserContact**

716 The Scoping Instance of an instance of CIM_UserContact shall be defined as follows:

- 717 1) From an instance of CIM_UserContact, traverse the CIM_AssignedIdentity association to locate
718 instances of CIM_Identity.
- 719 2) From each found CIM_Identity instance, traverse the CIM_ServiceAffectsElement association to
720 locate instances of CIM_AccountManagementService.

721 The Scoping Instance of the CIM_AccountManagementService shall be the Scoping Instance of the
722 CIM_UserContact instance.

723 **7.5 Managing Account Identity Groups**

724 Management of account identity groups on the managed system may be supported. This behavior is
725 optional. This section describes the requirements when this behavior is implemented.

726 **7.5.1 Managing Local Account Identity Groups**

727 Each instance of CIM_Group shall be associated with an instance of CIM_ComputerSystem through the
728 CIM_OwningCollectionElement association.

729 **7.5.2 Identity for a Group**

730 One or more instances of CIM_Identity shall be associated with an instance of CIM_Group through the
731 CIM_AssignedIdentity association.

732 **7.5.3 Relating an Account Identity to a Group**

733 CIM_Account may be grouped through its account identity (CIM_Identity) only. CIM_Account is
734 associated with CIM_Identity through the CIM_AssignedIdentity association. One or more instances of
735 CIM_Identity may be associated with an instance of CIM_Group through the CIM_MemberOfCollection
736 association.

737 If an instance of CIM_Group representing a group of account identities is implemented, then that instance
738 shall aggregate instances of CIM_Identity representing those identities via the CIM_MemberOfCollection
739 aggregation.

740 If an instance of CIM_Group representing a group of account identities is present, then that instance shall
741 be associated to the scoping CIM_ComputerSystem by an instance of CIM_OwningCollectionElement.

742 An instance of CIM_Account's identity shall be associated with an instance of CIM_Group only if the
743 CIM_ComputerSystem instance with which the CIM_Account instance is associated through an instance
744 of CIM_AccountOnSystem is the same CIM_ComputerSystem instance with which the CIM_Group
745 instance is associated through an instance of CIM_OwningCollectionElement.

746 **7.6 Representing Access Ingress Point**

747 For a particular instance of CIM_Identity, the ingress point through which a currently authenticated
748 session is being maintained may be identified by an **optional** instance of CIM_Dependency. Such an
749 ingress point may be a system, service, protocol endpoint, or other entity through which requests can
750 flow. An instance of CIM_Dependency between an instance of CIM_Identity and an instance of
751 CIM_ManagedElement shall not exist except to represent an authenticated session.

752 If instantiated, the instance of CIM_Dependency shall be implemented as specified in section 10.9.

753 **7.7 Identity Context**

754 A security principal, represented by an instance of CIM_Identity, may be scoped to one or more ingress
755 points by optional instances of CIM_IdentityContext. (Each ingress point may be a system, service,
756 protocol endpoint, or other entity through which requests can flow.)

757 The default ingress point for an instance of CIM_Identity is the CIM_System associated with the
758 CIM_AccountManagementService (via CIM_HostedService), that manages that instance of CIM_Identity
759 (as indicated by CIM_ServiceAffectsElement).

760 Unless otherwise specified by an instance of CIM_IdentityContext, the only allowed ingress point for
761 requests of a particular security principal shall be the default ingress point of the related CIM_Identity
762 instance.

763 If any instances of CIM_IdentityContext are associated to a particular CIM_Identity instance, then only
764 requests flowing through associated ingress points shall be allowed for the security principal represented
765 by that CIM_Identity.

766 NOTE 1: This association is many to many, indicating that the allowed request scope of a particular CIM_Identity
767 instance may be defined by several elements. However, it is likely that there will only be a single scoping instance,
768 which is likely to be the default specified above.

769 NOTE 2: The context of an instance of CIM_Identity has no effect on the scope of the privileges (if any) that are
770 granted to the represented security principal. Rather, the context provides information about when one security
771 principal versus another will be selected when credentials are provided that identify an authenticated entity.

772 **8 Methods**

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

775 **8.1 CIM_AccountManagementService.CreateAccount()**

776 The CIM_AccountManagementService.CreateAccount() method is used to create accounts on a
777 managed system. When the method returns a value of 0 (zero), a new instance of CIM_Account shall be
778 associated with the CIM_ComputerSystem instance that is identified by the System parameter through
779 the CIM_AccountOnSystem association such that the values of the properties of the instance of
780 CIM_Account are the values of the non-Null properties of the template account instance that is specified
781 by the AccountTemplate parameter. The value of the Account parameter shall be a reference to the
782 instance of CIM_Account. A newly created instance of CIM_Identity shall be associated with the

783 CIM_Account instance through the CIM_AssignedIdentity association. The instance of CIM_Identity shall
 784 be associated with the CIM_AccountManagementService through the CIM_ServiceAffectsElement
 785 association.

786 When the CIM_ComputerSystem instance identified by the System parameter is not associated with the
 787 CIM_AccountManagementService instance through the CIM_HostedService association, the method
 788 shall return the value 2.

789 CreateAccount() method return code values shall be as specified in Table 2. CreateAccount() method
 790 parameters are specified in Table 3.

791 No standard messages are defined for this method.

792 **Table 2 – CIM_AccountManagementService.CreateAccount() Method: Return Code Values**

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

793 **Table 3 – CIM_AccountManagementService.CreateAccount() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	System	CIM_ComputerSystem REF	Reference to scoping system
IN, EmbeddedInstance, REQ	AccountTemplate		Template for Account to create See section 8.1.1.
OUT	Account	CIM_Account REF	Reference to newly created Account
OUT	Identity	REF CIM_Identity	References to newly created Identity

794 **8.1.1 Account Template Requirements**

795 This section details the requirements for the AccountTemplate parameter.

796 When the AccountTemplate embedded instance contains the UserPasswordEncryptionAlgorithm property
 797 and the value specified for the property is not a supported value as defined in section 7.1.3.1 the method
 798 shall return the value 2.

799 When the AccountTemplate embedded instance contains the UserPassword property and the value
 800 specified for the property is not a supported value as defined in section 7.1.3.1 the method shall return
 801 the value 2.

802 When the AccountTemplate embedded instance contains the PasswordHistoryDepth property and the
 803 value specified for the property is not a supported value as defined in section 7.3.5, the method shall
 804 return the value 2.

805 When the AccountTemplate embedded instance contains the PasswordExpiration property and the value
 806 specified for the property is not a supported value as defined in section 7.3.5, the method shall return the
 807 value 2.

808 When the AccountTemplate embedded instance contains the ComplexPasswordRulesEnforced property
 809 and the value specified for the property is not a supported value as defined in section 7.3.5, the method
 810 shall return the value 2.

811 When the AccountTemplate embedded instance contains the InactivityTimeout property and the value
812 specified for the property is not a supported value as defined in section 7.3.5, the method shall return the
813 value 2.

814 When the AccountTemplate embedded instance contains the MaximumSuccessiveLoginFailures property
815 and the value specified for the property is not a supported value as defined in section 7.3.5, the method
816 shall return the value 2.

817 If the AccountTemplate embedded instance contains the LastLogin property, the value specified shall be
818 ignored.

819 **8.1.2 Account Default Values**

820 This section details how default values are supplied for instances of CIM_Account that are created by
821 using the CreateAccount() method.

822 **8.1.2.1 Using a Default Configuration**

823 When an instance of CIM_AccountSettingData is associated with the CIM_AccountManagementService
824 through the CIM_ElementSettingData association where the CIM_ElementSettingData.IsNext property
825 has the value 1 (Is Next), the requirements specified in this section shall be met.

826 For each non-Null property of the instance of CIM_AccountSettingData, if a value is not provided for the
827 corresponding property of the embedded instance specified by the AccountTemplate parameter, the
828 property of the instance of CIM_Account created by the method shall have the value of the property of the
829 CIM_AccountSettingData instance.

830 **8.1.2.2 Using Implicit Defaults**

831 When no instance of CIM_AccountSettingData is associated with the CIM_AccountManagementService
832 through the CIM_ElementSettingData association where the CIM_ElementSettingData.IsNext property
833 has the value 1 (Is Next), the requirements specified in this section shall be met.

834 For each non-Null property of the instance of CIM_AccountSettingData, if a value is not provided for the
835 corresponding property of the embedded instance specified by the AccountTemplate provider, the value
836 of the property of the instance of CIM_Account created by the method shall have an implementation-
837 specific value.

838 **8.1.3 CIM_AccountManagementService.CreateAccount() Conditional Support**

839 When the OperationsSupported property of the associated instance of
840 CIM_AccountManagementCapabilities contains the value 2 (Create), the
841 CIM_AccountManagementService.CreateAccount() method shall be implemented and shall not return a
842 value of 1 (Unsupported). When the OperationsSupported property of the associated instance of
843 CIM_AccountManagementCapabilities does not contain the value 2 (Create), the
844 CIM_AccountManagementService.CreateAccount() method may be implemented; if implemented, it shall
845 return a value of 1 (Operation unsupported).

846 **8.2 CIM_Account.RequestStateChange()**

847 Invoking the CIM_Account.RequestStateChange() method changes the element's state to the value
848 specified in the RequestedState parameter. The Enabled and Disabled values of the RequestedState
849 parameter correspond to enabling or disabling the functionality represented by the instance of
850 CIM_Account. A value of 2 (Enabled) shall correspond to a request to enable the account and place it in
851 the enabled state.

852 A value of 3 (Disabled) shall place the account in the disabled state.

- 853 A value of 6 (Offline) shall place the account into the offline state.
- 854 When the RequestedState parameter has the value 2 (Enabled), the method may return the value 2 if the
855 account is not properly configured.
- 856 See section 7.3.3.3 for information about the effect of this method on the RequestedState property.
- 857 The method shall be considered successful if the availability of the functionality upon completion of the
858 method corresponds to the desired availability indicated by the RequestedState parameter. It is not
859 necessary that an actual change in state occur for the method to be considered successful. It is sufficient
860 that the resultant state be equal to the requested state. Upon successful completion of the method, the
861 Return Value shall be 0 (zero).
- 862 See section 7.3.3.4 for information about the effect of this method on the EnabledState property.
- 863 RequestStateChange() method return code values shall be as specified in Table 4.
864 RequestStateChange() method parameters are specified in Table 5.
- 865 No standard messages are defined.
- 866 Invoking the CIM_Account.RequestStateChange() method multiple times could result in earlier requests
867 being overwritten or lost.

868 **Table 4 – CIM_Account.RequestStateChange() Method: Return Code Values**

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred
0x1000	Job started: REF returned to started CIM_ConcreteJob

869 **Table 5 – CIM_Account.RequestStateChange() Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values: 2 (Enabled) 3 (Disabled) 6 (Offline)
OUT	Job	CIM_ConcreteJob REF	Returned if job started
IN, REQ	TimeoutPeriod	datetime	Client-specified maximum amount of time the transition to a new state is supposed to take: 0 or NULL – No time requirements <interval> – Maximum time allowed

870 **8.2.1 CIM_Account.RequestStateChange() Conditional Support**

- 871 When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least
872 one value, the CIM_Account.RequestStateChange() method shall be implemented and supported. The
873 CIM_Account.RequestStateChange() method shall not return a value of 1 (Unsupported).

874 **8.3 Profile Conventions for Operations**

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

877 The default list of operations is as follows:

- 878 • GetInstance
- 879 • Associators
- 880 • AssociatorNames
- 881 • References
- 882 • ReferenceNames
- 883 • EnumerateInstances
- 884 • EnumerateInstanceNames

885 **8.4 CIM_Account**

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

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

890 **Table 6 – Operations: CIM_Account**

Operation	Requirement	Messages
GetInstance	Mandatory. See section 8.4.1.	None
ModifyInstance	Conditional. See section 8.4.2.	None
DeleteInstance	Conditional. See section 8.4.3.	None

891 **8.4.1 CIM_Account – GetInstance Operation**

892 The following are possible behaviors and are mutually exclusive:

- 893 • When the GetInstance operation is executed against an instance of CIM_Account and the
894 underlying account has a valid password, the value of the CIM_Account.UserPassword property
895 shall be an array of length zero to indicate that the account has a password configured and is
896 unable or unwilling to return the value in clear text.
- 897 • When the GetInstance operation is executed against an instance of CIM_Account and the
898 underlying account does not have a valid password, the CIM_Account.UserPassword property
899 shall be Null.

900 **8.4.2 CIM_Account – ModifyInstance Operation**

901 The ModifyInstance operation shall be supported if and only if the
902 OperationsSupported property contains the value 3 (Modify) for an instance of
903 CIM_AccountManagementCapabilities that is associated through the
904 CIM_ElementCapabilities association with an instance of
905 CIM_AccountManagementService associated through CIM_ServiceAffectsElement with an instance of
906 CIM_Identity that is associated with the instance of CIM_Account through CIM_AssignedIdentity.

907 As described in 7.1.3.1 the UserPassword property of CIM_Account may be in clear text or be encrypted.
 908 Encrypting UserPassword may be required since the network session may not be encrypted.

909 When the ModifyInstance operation is supported and a value is specified for the
 910 CIM_Account.UserPassword property and the CIM_Account.UserPasswordEncryptionAlgorithm property
 911 has no value or has the value 0 (None), the value of the CIM_Account.UserPassword property shall be
 912 clear text without encryption.

913 When the ModifyInstance operation is supported and a value is specified for the
 914 CIM_Account.UserPassword property and the CIM_Account.UserPasswordEncryptionAlgorithm property
 915 has a non-zero value, the value of the CIM_Account.UserPassword property shall be encrypted in the
 916 form specified by the value of the CIM_Account.UserPasswordEncryptionAlgorithm property

917 **8.4.3 CIM_Account – DeleteInstance Operation**

918 The DeleteInstance operation shall be supported if and only if the OperationsSupported property contains
 919 the value 4 (Delete) for an instance of CIM_AccountManagementCapabilities that is associated through
 920 the CIM_ElementCapabilities association with an instance of CIM_AccountManagementService
 921 associated through CIM_ServiceAffectsElement with an instance of CIM_Identity that is associated with
 922 the instance of CIM_Account through CIM_AssignedIdentity.

923 When the associated instance of CIM_Identity is not associated with any other instances of
 924 CIM_ManagedElement through the CIM_AssignedIdentity association, the CIM_Identity instance shall be
 925 deleted.

926 When the associated instance of CIM_EnabledLogicalElementCapabilities is not associated with any
 927 other instance of CIM_Account through the CIM_ElementCapabilities association, the instance of
 928 CIM_EnabledLogicalElementCapabilities shall be deleted.

929 Any association that references the instance of CIM_Account shall be deleted.

930 **8.5 CIM_EnabledLogicalElementCapabilities**

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

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

933 **8.6 CIM_AccountOnSystem**

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

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

938 **Table 7 – Operations: CIM_AccountOnSystem**

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

939 **8.7 CIM_AccountManagementCapabilities**

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

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

942 **8.8 CIM_AccountManagementService**

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

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

947 **Table 8 – Operations: CIM_AccountManagementService**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.8.1.	None

948 **8.8.1 CIM_AccountManagementService – ModifyInstance Operation**

949 This section details the specific requirements for the ModifyInstance operation applied to an instance of
 950 CIM_AccountManagementService.

951 **8.8.1.1 CIM_AccountManagementService.ElementName Property**

952 When an instance of CIM_AccountManagementCapabilities is associated with the
 953 CIM_AccountManagementService instance and the
 954 CIM_AccountManagementCapabilities.ElementNameEditSupported property has a value of TRUE, the
 955 implementation shall allow the ModifyInstance operation to change the value of the ElementName
 956 property of the CIM_AccountManagementService instance. The ModifyInstance operation shall enforce
 957 the length restriction specified in the MaxElementNameLen property of the
 958 CIM_AccountManagementCapabilities instance. The ModifyInstance operation shall enforce the regular
 959 expression specified in the ElementNameMask property of the CIM_EnabledLogicalElementCapabilities.

960 When an instance of CIM_AccountManagementCapabilities is not associated with the
 961 CIM_AccountManagementService instance, or the ElementNameEditSupported property of the
 962 CIM_AccountManagementCapabilities instance has a value of FALSE, the implementation shall not allow
 963 the ModifyInstance operation to change the value of the ElementName property of the
 964 CIM_AccountManagementService instance.

965 **8.9 CIM_AccountSettingData**

966 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
 967 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in
 968 the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

970 **Table 9 – Operations: CIM_AccountSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional	None

971 **8.10 CIM_AssignedIdentity**

972 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
 973 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations
 974 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

976 **Table 10 – Operations: CIM_AssignedIdentity**

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

977 **8.11 CIM_Dependency**

978 Table 11 lists implementation requirements for operations. If implemented, these operations shall be
 979 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations
 980 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

982 **Table 11 – Operations: CIM_Dependency**

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

983 **8.12 CIM_ElementCapabilities**

984 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
 985 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations
 986 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

988 **Table 12 – Operations: CIM_ElementCapabilities**

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

989 **8.13 CIM_ElementSettingData**

990 Table 13 lists implementation requirements for operations. If implemented, these operations shall be
 991 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 13, all operations
 992 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

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

994 **Table 13 – Operations: CIM_ElementSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.13.1.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

995 **8.13.1 CIM_ElementSettingData – ModifyInstance**

996 The behavior of the ModifyInstance operation varies depending on the property of the association that is
 997 modified and the instances that are referenced by the association instance. The ModifyInstance operation
 998 shall not allow the IsDefault property to be modified. The ModifyInstance operation shall not allow the
 999 IsCurrent property to be modified.

1000 When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next), the
 1001 ModifyInstance operation shall implement the following behavior:

- 1002 1) The ModifyInstance operation may find another instance of CIM_ElementSettingData that
 1003 associates an instance of CIM_AccountSettingData with the instance of
 1004 CIM_AccountManagementService that is referenced by the target instance of
 1005 CIM_ElementSettingData where the IsNext property has a value of 1 (Is Next).
- 1006 2) For the instance of CIM_ElementSettingData found, the ModifyInstance operation shall modify
 1007 the value of its IsNext property to have a value of 2 (Is Not Next).

1008 **8.14 CIM_Group**

1009 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1010 NOTE: Related profiles may define additional requirements on operations for the profile class.

1011 **8.15 CIM_HostedService**

1012 Table 14 lists implementation requirements for operations. If implemented, these operations shall be
 1013 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 14, all operations
 1014 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1015 NOTE: Related profiles may define additional requirements on operations for the profile class.

1016

Table 14 – Operations: CIM_HostedService

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

1017 8.16 CIM_Identity

1018 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1019 NOTE: Related profiles may define additional requirements on operations for the profile class.

1020 8.17 CIM_IdentityContext

1021 Table 15 lists implementation requirements for operations. If implemented, these operations shall be
 1022 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 15, all operations
 1023 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1024 NOTE: Related profiles may define additional requirements on operations for the profile class.

1025

Table 15 – Operations: CIM_IdentityContext

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

1026 8.18 CIM_MemberOfCollection

1027 Table 16 lists implementation requirements for operations. If implemented, these operations shall be
 1028 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 16, all operations
 1029 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1030 NOTE: Related profiles may define additional requirements on operations for the profile class.

1031

Table 16 – Operations: CIM_MemberOfCollection

Operation	Requirement	Messages
CreateInstance	Optional. See section 8.18.1.	None
DeleteInstance	Optional. See section 8.18.2.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

1032 **8.18.1 CIM_MemberOfCollection – CreateInstance**

1033 The CreateInstance operation may be supported for CIM_MemberOfCollection. When the CreateInstance
1034 operation is supported, the CreateInstance operation shall fail under the following conditions:

- 1035 • An instance of CIM_MemberOfCollection already associates the specified CIM_Identity with the
1036 CIM_Group.
- 1037 • The resultant instance of CIM_MemberOfCollection does not satisfy the constraints specified in
1038 sections 7.5.3 and 10.18.

1039 **8.18.2 CIM_MemberOfCollection – DeleteInstance**

1040 The DeleteInstance operation may be supported for CIM_MemberOfCollection when the instance is used
1041 to associate an instance of CIM_Identity with an instance of CIM_Group.

1042 **8.19 CIM_OwningCollectionElement**

1043 Table 17 lists implementation requirements for operations. If implemented, these operations shall be
1044 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 17, all operations
1045 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1046 NOTE: Related profiles may define additional requirements on operations for the profile class.

1047 **Table 17 – Operations: CIM_OwningCollectionElement**

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

1048 **8.20 CIM_ServiceAffectsElement**

1049 Table 18 lists implementation requirements for operations. If implemented, these operations shall be
1050 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 18, all operations
1051 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1052 NOTE: Related profiles may define additional requirements on operations for the profile class.

1053 **Table 18 – Operations: CIM_ServiceAffectsElement**

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

1054 **8.21 CIM_SettingsDefineCapabilities**

1055 Table 19 lists implementation requirements for operations. If implemented, these operations shall be
1056 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 19, all operations
1057 in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

1058 NOTE: Related profiles may define additional requirements on operations for the profile class.

1059

Table 19 – Operations: CIM_SettingsDefineCapabilities

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

1060 **8.22 CIM_UserContact**1061 All operations in the default list in 8.3 shall be implemented as defined in [DSP0200](#).

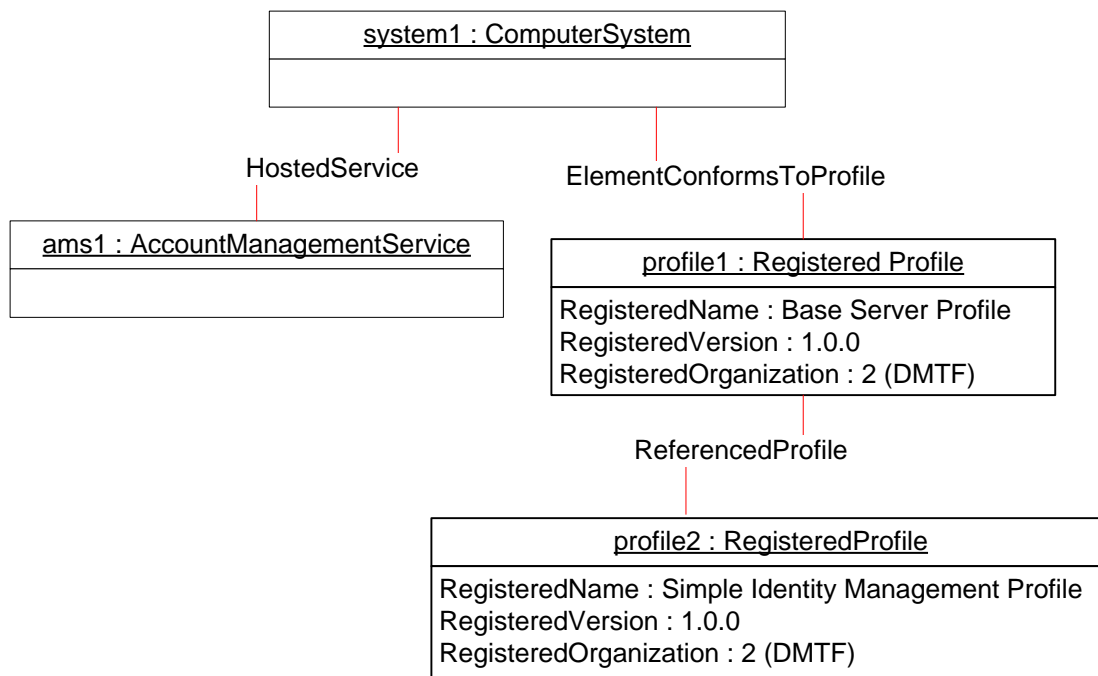
1062 NOTE: Related profiles may define additional requirements on operations for the profile class.

1063 **9 Use Cases**

1064 This section contains object diagrams and use cases for the *Simple Identity Management Profile*. The
 1065 contents of this section are for informative purposes only and do not constitute normative requirements
 1066 for implementations of this specification.

1067 **9.1 Profile Registration**

1068 Figure 2 describes one of the ways that the implementation can advertise the instantiation of the *Simple*
 1069 *Identity Management Profile*. Using scoping instance methodology as described in the [Profile Registration](#)
 1070 [Profile](#), profile2 contains the version information for the *Simple Identity Management Profile*
 1071 implementation.

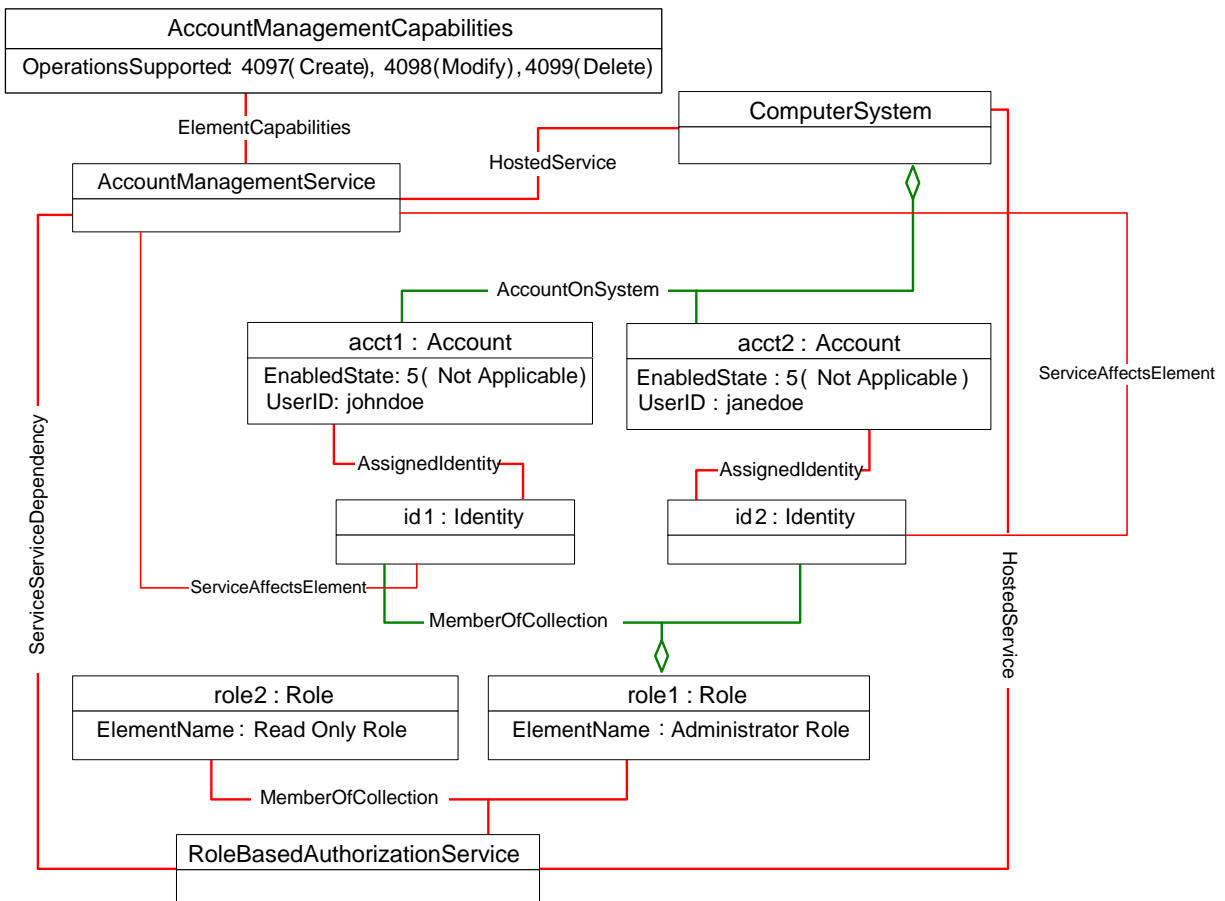


1072

1073

Figure 2 – Profile Registration

1074 Figure 3 shows a system that supports management of local accounts for authentication and
 1075 authorization. The modeled system supports creation, modification, and deletion of accounts. Privilege
 1076 management is performed through assignment to Roles.



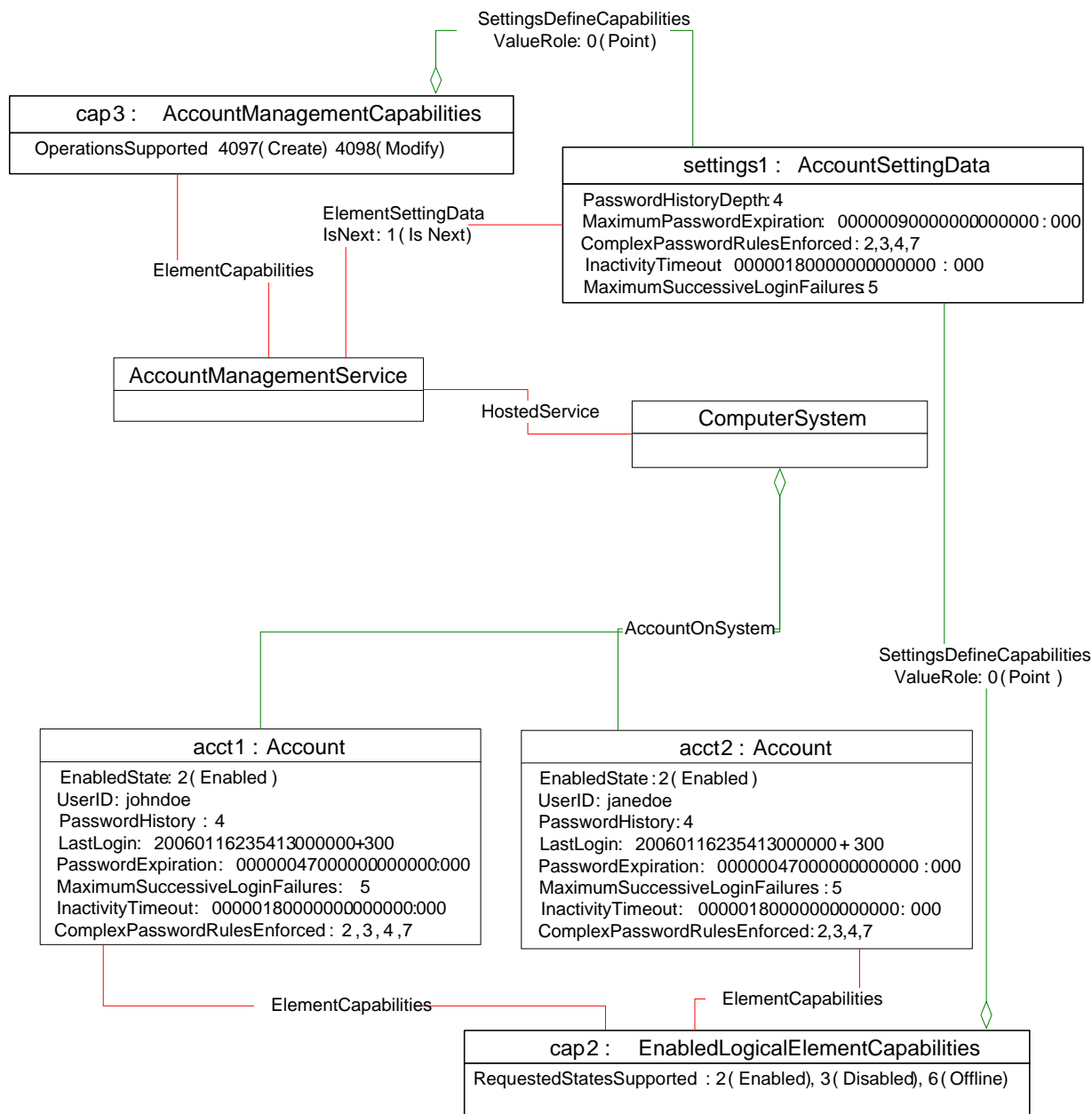
1077

1078

Figure 3 – Basic System Accounts

1079 Figure 4 shows a system that supports capabilities related to password management. Accounts created
 1080 through the CIM_AccountManagementService are required to maintain a history of the four previous
 1081 passwords, have the password changed every 90 days, enter a locked-out state after 180 days of
 1082 inactivity, and enter a locked-out state after five successive failed login attempts. Additionally, passwords
 1083 are required to have a minimum length, not contain the user ID, contain at least one numeric character,
 1084 and enforce a maximum number of repeating characters. These requirements are indicated by the
 1085 CIM_SettingsDefineCapabilities association between settings1 and cap3.

1086 acct1 and acct2 operate under the same password constraints as new accounts created through the
 1087 CIM_AccountManagementService. This behavior is indicated by the CIM_SettingsDefineCapabilities
 1088 association between cap2 and settings1. The password for each account is required to be changed every
 1089 90 days. Each account currently has 47 days until the password needs to be changed. Thus the
 1090 password for each account was last changed 43 days ago. Similarly, the accounts are required to enter a
 1091 locked-out state after 180 days of inactivity. Each account currently has 180 days until it will be locked.
 1092 Therefore each account was last accessed today.



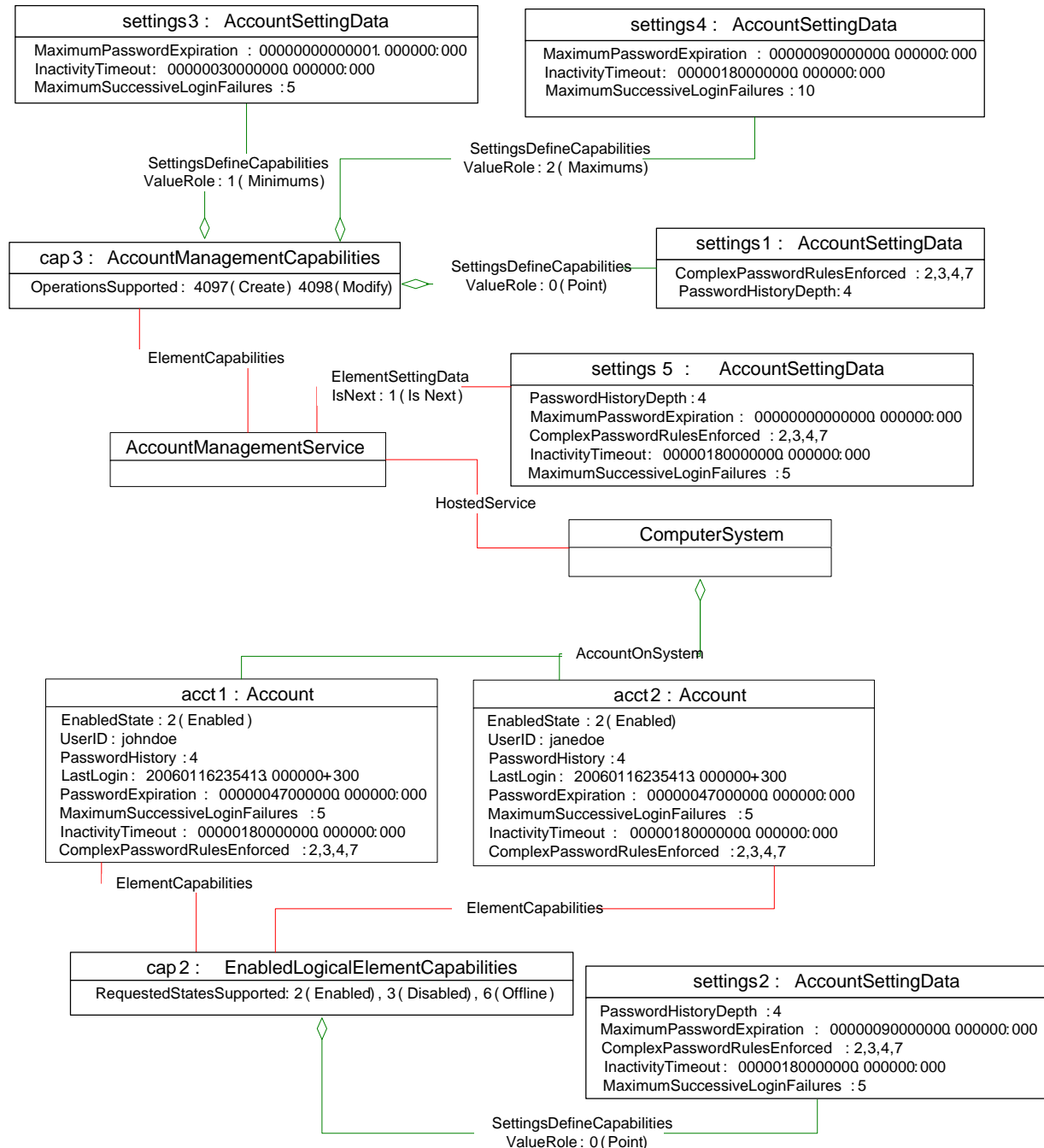
1093

1094

Figure 4 – Full Account Capabilities

1095 Figure 5 also shows a system that supports capabilities related to password management. Accounts
 1096 created through the CIM_AccountManagementService are required to maintain a history of the four
 1097 previous passwords. Account passwords are required to be changed at least every 90 days. The inactivity
 1098 timeout can be configured to be from 30 to 180 days. The number of successive failed login attempts can
 1099 be configured to be between five and ten. Additionally, passwords are required to have a minimum length,
 1100 not contain the user ID, contain at least one numeric character, and enforce a maximum number of
 1101 repeating characters. These constraints are indicated by the CIM_SettingsDefineCapabilities association
 1102 between cap3 and settings1, settings3, and settings4. acct1 and acct2 operate under the same password
 1103 constraints. These constraints are within the range allowed for created accounts. These constraints are
 1104 indicated by the CIM_SettingsDefineCapabilities association between cap2 and settings2. The password

1105 for each account is required to be changed every 90 days. Each account currently has 47 days until the
 1106 password needs to be changed. Thus, the password for each account was last changed 43 days ago.
 1107 Similarly, the accounts are required to enter a locked-out state after 180 days of inactivity. Each account
 1108 currently has 180 days until it will be locked. Therefore each account was last accessed today.
 1109 AccountSettingData settings5 shows the default setting.



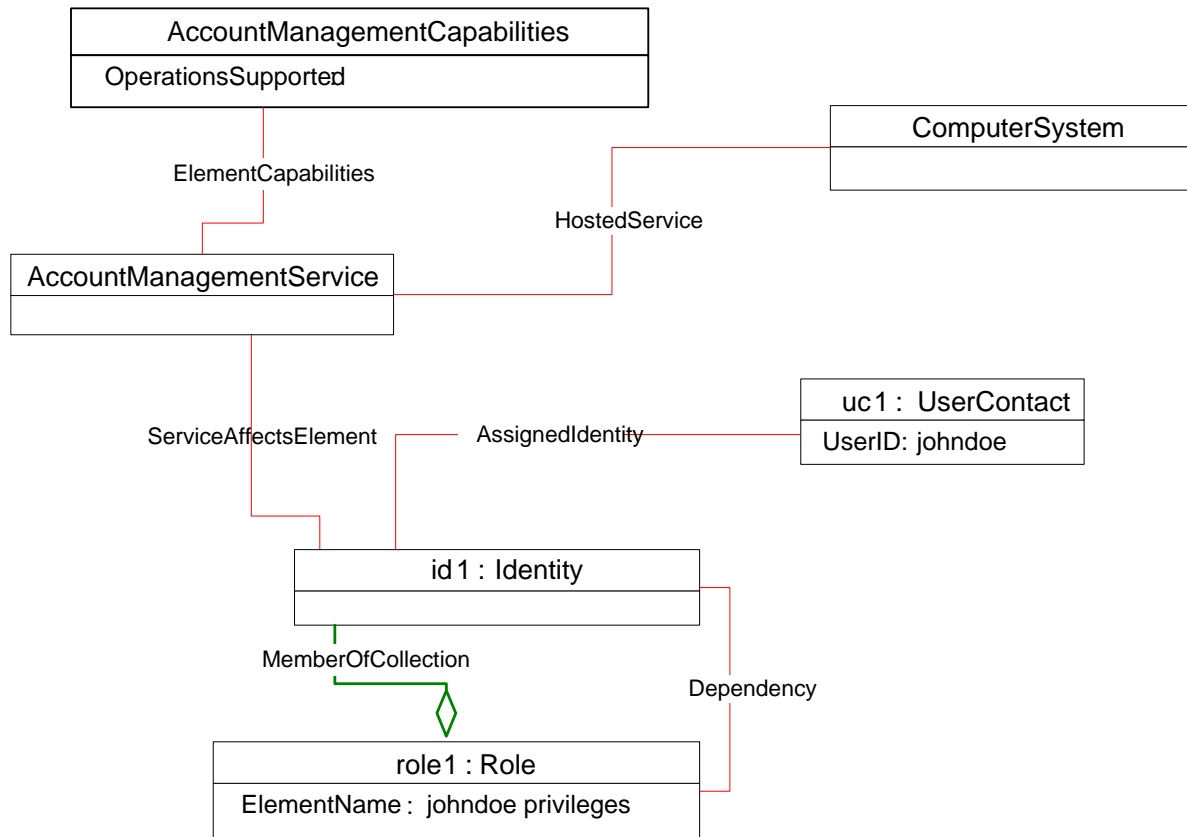
1110

1111

Figure 5 – Account Capabilities with Ranges

1112

1113 Figure 6 shows a system that has an active third-party authenticated user. The system does not have any
 1114 local accounts configured. The CIM_AccountManagementCapabilities.OperationsSupported property
 1115 indicates that account management is not supported. The user johndoe has the privileges specified by
 1116 role1.



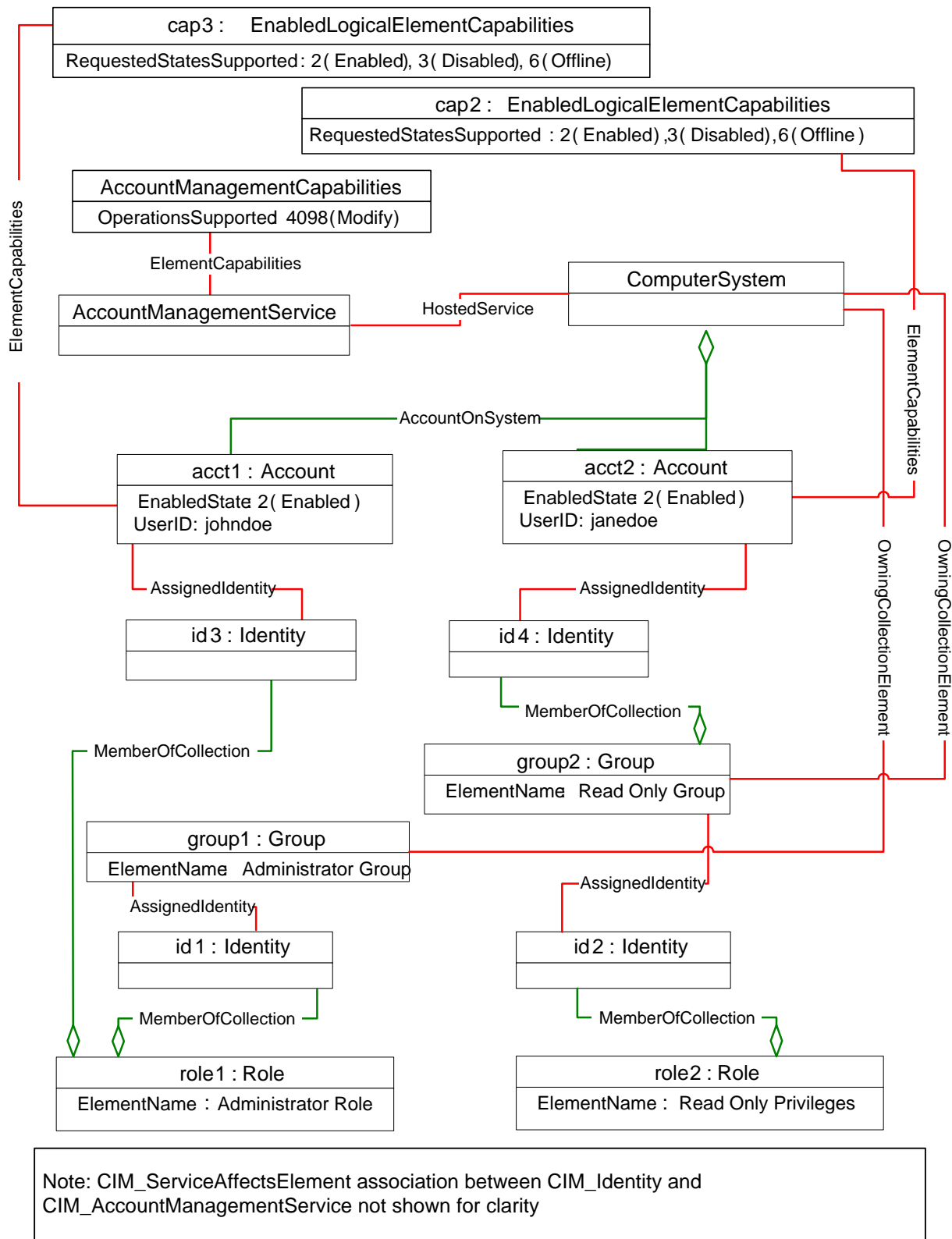
1117

1118

Figure 6 – Third-Party Authenticated User

1119 Figure 7 shows a system that supports Account Identity Groups. This object diagram has two groups:
 1120 group1 and group2. id1 and id2 represent the security principals for group1 and group2, respectively, as
 1121 indicated by the CIM_AssignedIdentity association instances. Two roles are supported by the system:
 1122 role1 and role2. This system has two local accounts: acct1 and acct2. The
 1123 CIM_AccountManagementCapabilities.OperationsSupported property indicates that account creation and
 1124 deletion are not supported. Therefore, these two accounts are fixed and the system does not support any
 1125 additional accounts. The accounts themselves can be enabled and disabled, as indicated by cap2 and
 1126 cap3. id3 and id4 represent the security principals for acct1 and acct2 respectively, as indicated by the
 1127 CIM_AssignedIdentity association instances.

1128 Privilege management for accounts and groups is managed directly through membership in a role. As
 1129 shown, acct1 is a member of role1 and therefore has the privileges of role1. acct2 is a member of group2
 1130 and inherits the privileges of role2.



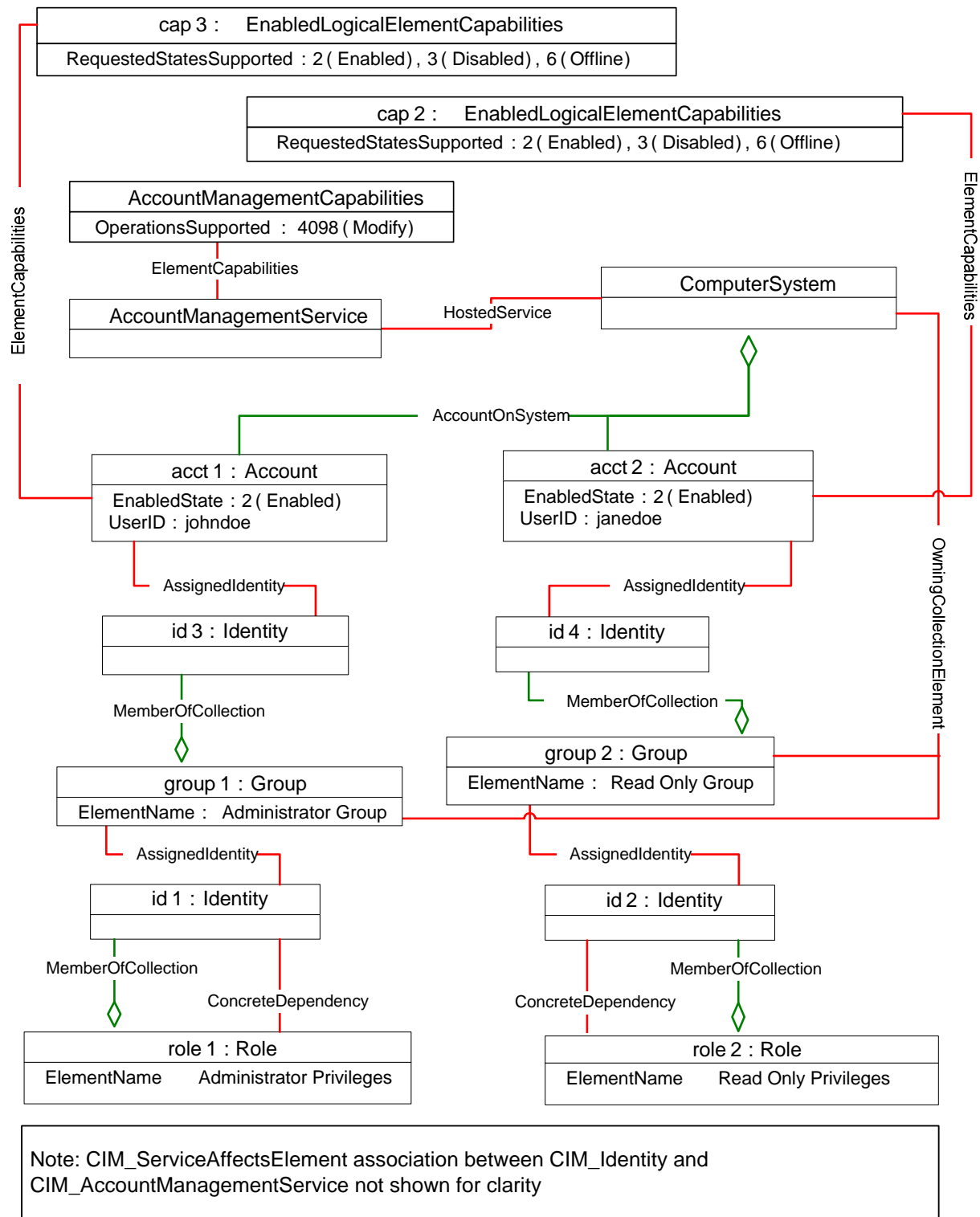
1131

1132

Figure 7 – Accounts with Group Membership

1133 Figure 8 shows a system that uses group membership to manage the privileges available to accounts.
1134 This object diagram has two groups: group1 and group2. id1 and id2 represent the security principals for
1135 group1 and group2, respectively, as indicated by the CIM_AssignedIdentity association instances. Two
1136 roles are supported by the system: role1 and role2. The roles are used to manage the capabilities of
1137 group1 and group2, respectively, as indicated by the CIM_Dependency association instances. This
1138 system has two local accounts: acct1 and acct2. The
1139 CIM_AccountManagementCapabilities.OperationsSupported property indicates that account
1140 management is not supported. Therefore these two accounts are fixed and the system does not support
1141 any additional accounts. The accounts themselves can be enabled and disabled, as indicated by cap2
1142 and cap3. id3 and id4 represent the security principals for acct1 and acct2, respectively, as indicated by
1143 the CIM_AssignedIdentity association instances.

1144 Privilege management for accounts is managed through membership in groups. The lack of CIM_Role
1145 instances that are not associated through CIM_Dependency to an instance of CIM_Identity that is
1146 associated to a CIM_Group results in the inability to assign a CIM_Account to a CIM_Role instance
1147 directly. acct1 is a member of group1 and therefore has the privileges of role1. acct2 is a member of
1148 group2 and therefore has the privileges of role2.

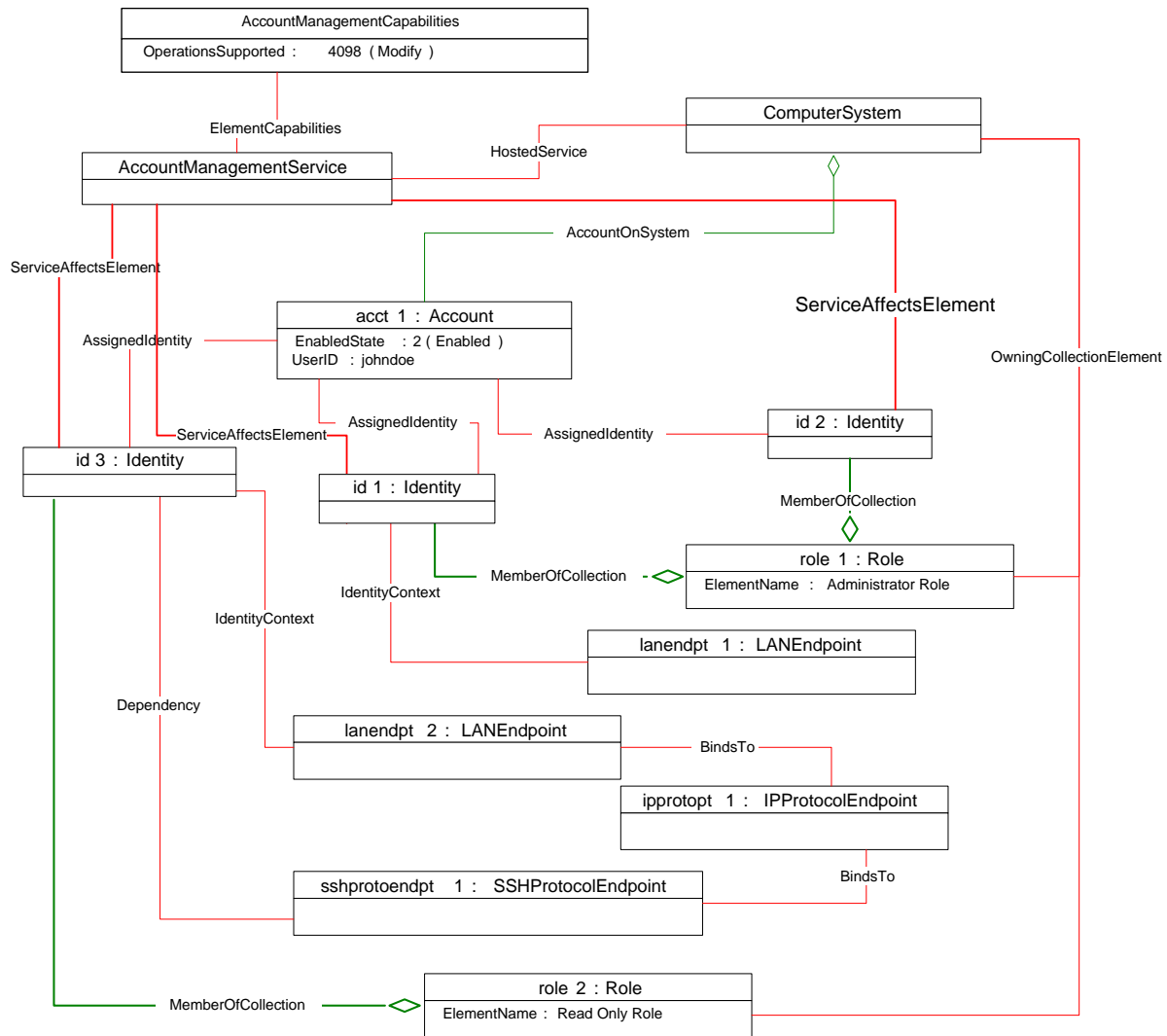


1149

1150

Figure 8 – Role-Oriented Groups

1151 Figure 9 shows a system with a local account where the privileges available to the account depend on the
 1152 mechanism through which the credentials are provided. The account has two security principals. Each
 1153 security principal is represented by an instance of CIM_Identity. id1 represents the security principal that
 1154 results from accessing the system over the network interface represented by landendpt1 using the
 1155 credentials of acct1. id3 represents the security principal that results from accessing the system over
 1156 landendpt2 using the credentials of acct1. id2 represents the security principal that results from accessing
 1157 the system using the credentials of acct1 through any other mechanism. In this system, accessing the
 1158 system over landendpt2 results in having the privileges of role2. Accessing the system any other way
 1159 results in having the privileges of role1 because id1 and id2 both belong to role1. The instance of
 1160 CIM_Dependency that associates sshprotoendpt1 and id3 indicates that the security principal whose
 1161 privileges were used for establishing the SSH session is id3.



1162

1163

Figure 9 – Access Ingress Point and Identity Context

1164 **9.2 Determine Whether CIM_Account.ElementName Can Be Modified**

1165 For a given instance of CIM_Account, a client can determine whether it can modify the ElementName as
1166 follows:

- 1167 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target
1168 instance.
- 1169 2) Query the value of the ElementNameEditSupported property of the
1170 CIM_EnabledLogicalElementCapabilities instance.

1171 If the value is TRUE, the client can modify the ElementName property of the target instance.

1172 **9.3 Determine Whether Account State Management Is Supported**

1173 For a given instance of CIM_Account, a client can determine whether state management is supported as
1174 follows:

- 1175 1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the
1176 CIM_Account instance.
- 1177 2) Query the value of the RequestedStatesSupported property.

1178 If at least one value is specified, state management is supported.

1179 **9.4 Determine Whether Account Management Is Supported**

1180 A client can determine if account management is supported for a system as follows:

- 1181 1) Starting at the CIM_ComputerSystem instance for the managed system, look for an instance of
1182 CIM_AccountManagementService with which it is associated through the CIM_HostedService
1183 association.
- 1184 2) Find an instance of CIM_AccountManagementCapabilities that is associated with the
1185 CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
- 1186 3) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

1187 If at least one value is contained in the array, account management is supported.

1188 **9.5 Create an Account**

1189 A client can create an account on a system as follows:

- 1190 1) Determine if account creation is supported as follows:
 - 1191 a) Starting at the CIM_ComputerSystem instance for the managed system, look for an
1192 instance of CIM_AccountManagementService with which it is associated through the
1193 CIM_HostedService association.
 - 1194 b) Find an instance of CIM_AccountManagementCapabilities that is associated with the
1195 CIM_AccountManagementService instance through the CIM_ElementCapabilities
1196 association.
 - 1197 c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported
1198 property.

1199 If the value 2 (Create) is contained in the array, account creation is supported.

- 1200 2) Create a template instance of CIM_Account.
- 1201 3) Invoke the CIM_AccountManagementService.CreateAccount() method, specifying the template
1202 instance.

1203 If the method returns a value of 0, the account has been successfully created.

1204 **9.6 Determine Account Defaults**

1205 A client can determine the default configuration for a newly created account as follows:

- 1206 1) Starting with the CIM_AccountManagementService, look for an instance of
1207 CIM_AccountSettingData with which it is associated through the CIM_ElementSettingData
1208 association where the CIM_ElementSettingData.IsNext property has the value 1 (Is Next).
- 1209 2) If an instance is found, query the values of the properties to determine the default configuration.

1210 If an instance is not found, the default values are indeterminate.

1211 **9.7 Delete an Account**

1212 A client can delete an account on a system as follows:

- 1213 1) Determine if account deletion is supported as follows:
 - 1214 a) Starting at the CIM_Account instance, look for an instance of
1215 CIM_AccountManagementService with which it is associated. CIM_Account is associated
1216 with CIM_Identity through the CIM_AssignedIdentity association and CIM_Identity is
1217 associated with the AccountManagementService through the CIM_ServiceAffectsElement
1218 association
 - 1219 b) Find an instance of CIM_AccountManagementCapabilities that is associated with the
1220 CIM_AccountManagementService instance through the CIM_ElementCapabilities
1221 association.
 - 1222 c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported
1223 property.

1224 If the value 4 (Delete) is contained in the array, account deletion is supported.

- 1225 2) Invoke the DeleteInstance operation against the instance of CIM_Account.

1226 **9.8 Modify the Password for an Account**

1227 A client can modify the password for an account on a system as follows:

- 1228 1) Determine if account modification is supported as follows:
 - 1229 a) Starting at the CIM_Account instance, look for an instance of
1230 CIM_AccountManagementService with which it is associated. CIM_Account is associated
1231 with CIM_Identity through the CIM_AssignedIdentity association and CIM_Identity is
1232 associated with the AccountManagementService through the CIM_ServiceAffectsElement
1233 association
 - 1234 b) Find an instance of CIM_AccountManagementCapabilities that is associated with the
1235 CIM_AccountManagementService instance through the CIM_ElementCapabilities
1236 association.
 - 1237 c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported
1238 property.

1239 If the value 3 (Modify) is contained in the array, account modification is supported.

- 1240 2) Invoke the GetInstance operation against the target instance of CIM_Account
- 1241 3) Modify the UserPassword property.
- 1242 4) Invoke the ModifyInstance operation.

1243 9.9 Clear an Account

1244 A client can clear an account as follows:

- 1245 1) Starting at the instance of `CIM_Account`, look for an instance of
1246 `CIM_EnabledLogicalElementCapabilities` with which it is associated through the
1247 `CIM_ElementCapabilities` association.
- 1248 2) If an instance is found, query the `RequestedStatesSupported` property to determine if it contains
1249 the value 3 (Disabled).
- 1250 3) Invoke the `CIM_Account.RequestStateChange()` method specifying a value of 3 (Disabled).

1251 9.10 Change State to Enabled Offline

1252 A client can change state to Enabled Offline an account as follows:

- 1253 1) Starting at the instance of `CIM_Account`, look for an instance of
1254 `CIM_EnabledLogicalElementCapabilities` with which it is associated through the
1255 `CIM_ElementCapabilities` association.
- 1256 2) If an instance is found, query the `RequestedStatesSupported` property to determine if it contains
1257 the value 6 (Enabled but Offline).
- 1258 3) Invoke the `CIM_Account.RequestStateChange()` method specifying a value of 6 (Enabled but
1259 Offline).

1260 9.11 Add an Account Identity to a Group

1261 A client can add an account identity to a group as follows:

- 1262 1) Find an instance of `CIM_Identity` that is associated with the target instance of `CIM_Account`
1263 through the `CIM_AssignedIdentity` association.
- 1264 2) Invoke the `CreateInstance` operation against `CIM_MemberOfCollection` where the template
1265 instance references the desired instances of `CIM_Identity` and `CIM_Group`.

1266 9.12 Remove an Account Identity from a Group

1267 A client can remove an account identity from a group as follows:

- 1268 1) Find each instance of `CIM_Identity` that is associated with the target `CIM_Account` instance
1269 through the `CIM_AssignedIdentity` association.
- 1270 2) For each instance of `CIM_Identity`, test whether it is associated with the target `CIM_Group`
1271 instance through the `CIM_MemberOfCollection` association.
- 1272 3) If the instance of `CIM_MemberOfCollection` exists, execute the `DeleteInstance` operation
1273 against it.

1274 9.13 Determine the Context of a Security Principal

1275 A client can determine the context of an instance of `CIM_Identity` by looking for one or more instances of
1276 `CIM_IdentityContext` that reference the targeted instance of `CIM_Identity`. If one or more instances are
1277 found, each referenced instance of `CIM_ManagedElement` provides context where the security principal
1278 will be used. Otherwise, the context of the `CIM_Identity` instance is the scope of the
1279 `CIM_ManagedElement` to which it is associated through `CIM_AssignedIdentity`.

1280 **10 CIM Elements**

1281 Table 20 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 1282 implemented as described in Table 20. Sections 7 (“Implementation”) and 8 (“Methods”) may impose
 1283 additional requirements on these elements.

1284

1285

Table 20 – CIM Elements: *Simple Identity Management Profile*

Element Name	Requirement	Description
Classes		
CIM_Account	Conditional	See sections 7.1.3 and 10.1.
CIM_AccountManagementCapabilities	Mandatory	See section 10.2.
CIM_AccountManagementService	Mandatory	See section 10.3.
CIM_AccountOnSystem	Conditional	See sections 7.1.3 and 10.4.
CIM_AccountSettingData	Optional	See section 10.5.
CIM_AssignedIdentity (CIM_Account)	Conditional	See sections 7.1.3 and 10.6.
CIM_AssignedIdentity (CIM_Group)	Optional	See sections 7.5.2 and 10.7.
CIM_AssignedIdentity (CIM_UserContact)	Optional	See sections 7.4.1 and 10.8.
CIM_Dependency	Optional	See section 10.9.
CIM_ElementCapabilities	Mandatory	See section 10.10.
CIM_ElementCapabilities	Optional	See sections 7.3.2 and 10.11.
CIM_ElementSettingData	Optional	See section 10.12.
CIM_EnabledLogicalElementCapabilities	Optional	See section 10.13.
CIM_Group	Optional	See section 10.14.
CIM_HostedService	Mandatory	See section 10.15.
CIM_Identity	Mandatory	See sections 7.1 and 10.16.
CIM_IdentityContext	Optional	See section 10.17.
CIM_MemberOfCollection	Optional	See sections 7.5.3 and 10.18.
CIM_OwningCollectionElement	Optional	See section 7.5.3 and 10.19.
CIM_RegisteredProfile	Mandatory	See section 10.24.
CIM_ServiceAffectsElement	Mandatory	See section 10.20.
CIM_SettingsDefineCapabilities	Optional	See section 10.21 and 10.22.
CIM_UserContact	Optional	See section 10.23.
Indications		
None defined in this profile		

1286 **10.1 CIM_Account**

1287 Table 21 details the requirements for instances of CIM_Account.

1288 **Table 21 – Class: CIM_Account**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
UserID	Mandatory	(pattern ".**")
UserPassword	Mandatory	(pattern ".**")
OrganizationName	Mandatory	(pattern ".**")
ElementName	Mandatory	See section 7.3.4.1.
UserPasswordEncryptionAlgorithm	Optional	See section 7.1.3.1.
OtherUserPasswordEncryptionAlgorithm	Conditional	Mandatory when UserPasswordEncryptionAlgorithm is 1 (Other).
PasswordHistoryDepth	Optional	See section 7.3.5.1.
PasswordExpiration	Optional	See section 7.3.5.2.
ComplexPasswordRulesEnforced	Optional	See section 7.3.5.3.
InactivityTimeout	Optional	See section 7.3.5.4.
MaximumSuccessiveLoginFailures	Optional	See section 7.3.5.5.
RequestedState	Mandatory	See section 7.3.3.3.
EnabledState	Mandatory	See section 7.3.3.4.
RequestStateChange()	Conditional	See section 7.3.3.2.

1289 **10.2 CIM_AccountManagementCapabilities**

1290 CIM_AccountManagementCapabilities indicates support for managing the account with which the service
 1291 is associated and indicates supported operations. Table 22 details the requirements for instances of
 1292 CIM_AccountManagementCapabilities.

1293 **Table 22 – Class: CIM_AccountManagementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	None
ElementNameEditSupported	Mandatory	See section 7.3.4.2.1.
MaxElementNameLen	Conditional	See section 7.3.4.2.2.
ElementName	Mandatory	pattern ".**"
OperationsSupported	Mandatory	None
SupportedUserPasswordEncryptionAlgorithms[]	Optional	See section 7.1.2.

1294 **10.3 CIM_AccountManagementService**

1295 Table 23 details the requirements for instances of CIM_AccountManagementService.

1296 **Table 23 – Class: CIM_AccountManagementService**

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
RequestedState	Mandatory	Matches 12 (Not Applicable)
EnabledState	Mandatory	Matches 2 (Enabled)
ElementName	Mandatory	See section 7.3.4.
CreateAccount()	Conditional	See section 8.1.

1297 **10.4 CIM_AccountOnSystem**

1298 Table 24 details the requirements for instances of CIM_AccountOnSystem.

1299 **Table 24 – Class: CIM_AccountOnSystem**

Elements	Requirement	Notes
GroupComponent	Mandatory	This property shall be a reference to CIM_ComputerSystem. Cardinality 1
PartComponent	Mandatory	This property shall be a reference to an instance of CIM_Account. Cardinality *

1300 **10.5 CIM_AccountSettingData**

1301 Table 25 details the requirements for instances of CIM_AccountSettingData.

1302 **Table 25 – Class: CIM_AccountSettingData**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
PasswordHistoryDepth	Optional	See section 7.3.5.1.
MaximumPasswordExpiration	Optional	See section 7.3.5.2.
ComplexPasswordRulesEnforced	Optional	See section 7.3.5.3.
InactivityTimeout	Optional	See section 7.3.5.4.
MaximumSuccessiveLoginFailures	Optional	See section 7.3.5.5.

1303 **10.6 CIM_AssignedIdentity (CIM_Account)**

1304 Table 26 details the requirements for instances of CIM_AssignedIdentity.

1305 **Table 26 – Class: CIM_AssignedIdentity (CIM_Account)**

Elements	Requirement	Notes
IdentityInfo	Mandatory	This property shall be a reference to CIM_Identity. Cardinality 1..*
ManagedElement	Mandatory	This property shall be a reference to CIM_Account. Cardinality *

1306 **10.7 CIM_AssignedIdentity (Group)**

1307 Table 27 details the requirements for instances of CIM_AssignedIdentity.

1308 **Table 27 – Class: CIM_AssignedIdentity (Group)**

Elements	Requirement	Notes
IdentityInfo	Mandatory	This property shall be a reference to CIM_Identity. Cardinality 1..*
ManagedElement	Mandatory	This property shall be a reference to CIM_Group. Cardinality 0..1

1309 **10.8 CIM_AssignedIdentity (UserContact)**

1310 Table 28 details the requirements for instances of CIM_AssignedIdentity.

1311 **Table 28 – Class: CIM_AssignedIdentity (UserContact)**

Elements	Requirement	Notes
IdentityInfo	Mandatory	This property shall be a reference to CIM_Identity. Cardinality 1..*
ManagedElement	Mandatory	This property shall be a reference to CIM_UserContact. Cardinality 0..1

1312 **10.9 CIM_Dependency (Access Ingress)**

1313 Table 29 details the requirements for instances of CIM_Dependency. CIM_Dependency is used to
1314 associate an instance of CIM_Identity with an instance of CIM_ManagedElement.

1315 **Table 29 – Class: CIM_Dependency (Access Ingress)**

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

1316 **10.10 CIM_ElementCapabilities (CIM_AccountManagementService)**

1317 CIM_ElementCapabilities associates an instance of CIM_AccountManagementCapabilities with the
 1318 Central Instance. Table 30 details the requirements for instances of CIM_ElementCapabilities.

1319 **Table 30 – Class: CIM_ElementCapabilities (CIM_AccountManagementService)**

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to the Central Instance. Cardinality 1..*
Capabilities	Mandatory	This property shall be a reference to an instance of CIM_AccountManagementCapabilities. Cardinality 1

1320 **10.11 CIM_ElementCapabilities (CIM_Account)**

1321 CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with an
 1322 instance of CIM_Account. Table 31 details the requirements for instances of CIM_ElementCapabilities.

1323 **Table 31 – Class: CIM_ElementCapabilities (CIM_Account)**

Elements	Requirement	Notes
ManagedElement	Mandatory	This property shall be a reference to CIM_Account. Cardinality *
Capabilities	Mandatory	This property shall be a reference to an instance of CIM_EnabledLogicalElementCapabilities. Cardinality 0..1

1324 **10.12 CIM_ElementSettingData**

1325 CIM_ElementSettingData associates instances of CIM_AccountSettingData with an
 1326 CIM_AccountManagementService instance. Table 32 details the requirements for instances of
 1327 CIM_ElementSettingData.

1328 **Table 32 – Class: CIM_ElementSettingData**

Elements	Requirement	Notes
ManagedElement	Mandatory	Key This property shall be a reference to the Central Instance AccountManagementService Cardinality *
SettingData	Mandatory	Key This property shall be a reference to an instance of CIM_AccountSettingData. Cardinality *
IsNext	Mandatory	Matches 1 (Is Next) or 2 (Is Not Next)

1329 **10.13 CIM_EnabledLogicalElementCapabilities**

1330 CIM_EnabledLogicalElementCapabilities indicates support for managing the state of the service as well
 1331 as the accounts with which the service is associated. Table 33 details the requirements for instances of
 1332 CIM_EnabledLogicalElementCapabilities.

1333 **Table 33 – Class: CIM_EnabledLogicalElementCapabilities**

Elements	Requirement	Notes
InstanceID	Mandatory	None
ElementName	Mandatory	pattern ".*"
RequestedStatesSupported	Mandatory	See section 7.3.3.5.
ElementNameEditSupported	Mandatory	See section 7.3.4.2.1.
MaxElementNameLen	Conditional	See section 7.3.4.2.2.
ElementNameMask	Conditional	See section 7.3.4.2.3

1334 **10.14 CIM_Group**

1335 Table 34 details the requirements for instances of CIM_Group.

1336 **Table 34 – Class: CIM_Group**

Elements	Requirement	Notes
CreationClassName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	pattern ".*"

1337 **10.15 CIM_HostedService**

1338 Table 35 details the requirements for instances of CIM_HostedService.

1339 **Table 35 – Class: CIM_HostedService**

Elements	Requirement	Notes
Antecedent	Mandatory	Key This property shall be a reference to the Scoping Instance. Cardinality 1
Dependent	Mandatory	Key This property shall be a reference to the Central Instance. Cardinality 1..*

1340 **10.16 CIM_Identity**

1341 Table 36 details the requirements for instances of CIM_Identity.

1342 **Table 36 – Class: CIM_Identity**

Elements	Requirement	Notes
InstanceID	Mandatory	Key
ElementName	Mandatory	pattern ".*"

1343 **10.17 CIM_IdentityContext**

1344 Table 37 details the requirements for instances of CIM_IdentityContext.

1345 **Table 37 – Class: CIM_IdentityContext**

Elements	Requirement	Notes
ElementInContext	Mandatory	This property shall be a reference to CIM_Identity. Cardinality *
ElementProvidingContext	Mandatory	This property shall be a reference to CIM_ManagedElement. Cardinality *

1346 **10.18 CIM_MemberOfCollection (Group Membership)**1347 Table 38 details the requirements for instances of CIM_MemberOfCollection when it is used to associate
1348 instances of CIM_Identity with instances of CIM_Group.1349 **Table 38 – Class: CIM_MemberOfCollection (Group Membership)**

Elements	Requirement	Notes
Collection	Mandatory	The value of this property shall be an instance of CIM_Group. Cardinality 0..1
Member	Mandatory	This property shall be a reference to an instance of CIM_Identity Cardinality 1..*

1350 **10.19 CIM_OwningCollectionElement**

1351 Table 39 details the requirements for instances of CIM_OwningCollectionElement.

1352 **Table 39 – Class: CIM_OwningCollectionElement**

Elements	Requirement	Notes
OwningElement	Mandatory	The value of this property shall be the Scoping Instance of this profile. Cardinality 1
OwnedElement	Mandatory	The value of this property shall be an instance of CIM_Group. Cardinality *

1353 **10.20 CIM_ServiceAffectsElement**

1354 Table 40 details the requirements for instances of CIM_ServiceAffectsElement.

1355 **Table 40 – Class: CIM_ServiceAffectsElement (Account)**

Elements	Requirement	Notes
AffectingElement	Mandatory	Key This property shall be a reference to the Central Instance of the profile. Cardinality 1
AffectedElement	Mandatory	Key This property shall be a reference to CIM_Identity. Cardinality *
ElementEffects	Mandatory	Matches 5 (Manages)

1356 **10.21 CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)**

1357 Table 41 details the requirements for instances of CIM_SettingsDefineCapabilities when it is used to
 1358 associate an instance of CIM_AccountSettingData with an instance of
 1359 CIM_AccountManagementCapabilities. The value of the PropertyPolicy property is fixed at 0
 1360 (Independent), which indicates that the value of each property on the referenced
 1361 CIM_AccountSettingData instances is independent of the values of the other properties. The ValueRole[]
 1362 property is fixed at the value 3 (Supported), which indicates that the value of each property on a
 1363 referenced instance of CIM_AccountSettingData represents an inclusive constraint.

1364 **Table 41 – Class: CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)**

Elements	Requirement	Notes
GroupComponent	Mandatory	Key This property shall be a reference to an instance of CIM_AccountManagementCapabilities. Cardinality 0..1
PartComponent	Mandatory	Key This property shall be a reference to CIM_AccountSettingData. Cardinality *
PropertyPolicy	Mandatory	Matches 0 (Independent)
ValueRole	Mandatory	Matches 3 (Supported)
ValueRange	Mandatory	Matches 0 (Point) or 1 (Minimums) or 2 (Maximums)

1365 **10.22 CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)**

1366 Table 42 details the requirements for instances of CIM_SettingsDefineCapabilities when it is used to
 1367 associate an instance of CIM_AccountSettingData with an instance of
 1368 CIM_EnabledLogicalElementCapabilities. The value of the PropertyPolicy property is fixed at 0
 1369 (Independent), which indicates that the value of each property on the referenced
 1370 CIM_AccountSettingData instances is independent of the values of the other properties. The ValueRole[]
 1371 property is fixed at the value 3 (Supported), which indicates that the value of each property on a
 1372 referenced instance of CIM_AccountSettingData represents an inclusive constraint.

1373 **Table 42 – Class: CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)**

Elements	Requirement	Notes
GroupComponent	Mandatory	Key This property shall be a reference to an instance of CIM_EnabledLogicalElementCapabilities. Cardinality *
PartComponent	Mandatory	Key This property shall be a reference to CIM_AccountSettingData. Cardinality *
PropertyPolicy	Mandatory	Matches 0 (Independent)
ValueRole	Mandatory	Matches 3 (Supported)
ValueRange	Mandatory	Matches 0 (Point) or 1 (Minimums) or 2 (Maximums)

1374 **10.23 CIM_UserContact**

1375 Table 43 details the requirements for instances of CIM_UserContact.

1376 **Table 43 – Class: CIM_UserContact**

Elements	Requirement	Notes
CreationClassName	Mandatory	Key
Name	Mandatory	Key
UserID	Mandatory	pattern ".**"
ElementName	Mandatory	pattern ".**"

1377 **10.24 CIM_RegisteredProfile**

1378 CIM_RegisteredProfile identifies the *Simple Identity Management Profile*. The CIM_RegisteredProfile
 1379 class is defined by the [DSP1033](#). With the exception of the mandatory values specified for the properties
 1380 in Table 44, the behavior of the CIM_RegisteredProfile instance is defined by the [DSP1033](#).

1381 **Table 44 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	Matches "Simple Identity Management"
RegisteredVersion	Mandatory	Matches "1.0.1"
RegisteredOrganization	Mandatory	Matches 2 ("DMTF")

1382

ANNEX A
(informative)**Change Log**

Version	Date	Author	Description
1.0.0a	10/30.2006	Aaron Merkin	Preliminary Standard
1.0.1	06/17/2009	Hemal Shah	Erratum version based on resolutions to comments on 1.0.0.
1.0.1	06/17/2009		DMTF Standard

1383
1384
1385
1386
1387

1388