



1
2
3
4

Document Number: DSP1106

Date: 2011-09-16

Version: 1.0.0

5 **Integrated Access Control Policy Management**
6 **Profile**

7 **Document Type: Specification**
8 **Document Status: DMTF Standard**
9 **Document Language: en-US**

10 Copyright notice

11 Copyright © 2011 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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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>.

32

Contents

34	Foreword	6
35	Introduction	7
36	1 Scope	9
37	2 Normative references	9
38	3 Terms and definitions	10
39	4 Symbols and abbreviated terms	11
40	5 Synopsis	11
41	6 Description	12
42	6.1 Access control policy composition	12
43	6.2 Access control policy distribution	14
44	7 Implementation	16
45	7.1 Access control service	16
46	7.2 Principals	17
47	7.3 Resources and resource-related capabilities	17
48	7.4 Policy transfer service access point	19
49	7.5 Access control policy	19
50	7.6 Policy activation service	20
51	8 Methods	20
52	8.1 Extrinsic methods	20
53	8.2 Profile conventions for operations	22
54	9 Use cases	26
55	9.1 Discover conformant access control service	26
56	9.2 Determine the principal type and the resource type	27
57	9.3 Determine the resource related capabilities	28
58	9.4 Compose access control policies	29
59	9.5 Determine policy transfer service	30
60	9.6 Distribute access control policies	31
61	9.7 Activate access policies	32
62	9.8 Deactivate access policies	33
63	10 CIM Elements	34
64	10.1 CIM_HostedService	35
65	10.2 CIM_AssociatedPolicyActivationService	35
66	10.3 CIM_ElementSettingData	35
67	10.4 CIM_ElementCapabilities	36
68	10.5 CIM_PolicySetComponent	36
69	10.6 CIM_ReusablePolicyContainer	36
70	10.7 CIM_ReusablePolicy	36
71	10.8 CIM_ServiceServiceDependency	37
72	10.9 CIM_ServiceAffectsElement	37
73	10.10 CIM_AccessControlService	37
74	10.11 CIM_RegisteredProfile	38
75	10.12 CIM_AccessControlServiceSettingData	38
76	10.13 CIM_PolicyActivationService	38
77	10.14 CIM_PolicyTransferServiceAccessPoint	39
78	10.15 CIM_AccessControlPolicyGroup	39
79	10.16 CIM_AccessControlPolicy	39
80	10.17 CIM_FileSystemCapabilities	40
81	10.18 CIM_RelationalDatabaseCapabilities	40
82	10.19 CIM_DatabaseContainsTable	40
83	ANNEX A (informative) Change Log	42

84
85
86
87
88
89
90
91
92
93
94
95
96
97

Figures

Figure 1 – Access control policy composition: Class diagram..... 13
 Figure 2 – Access control policy distribution: Class diagram..... 15
 Figure 3 – Supported resources and capabilities 18
 Figure 4 – IAM registered profile and access control service 27
 Figure 5 – Hosted file system and related capabilities 28
 Figure 6 – Accounts and resources for policy composition 29
 Figure 7 – Policy transfer service..... 30
 Figure 8 – Distributed access policies 31
 Figure 9 – Activated access policies..... 32
 Figure 10 – Flow diagram of the policy activation process 33
 Figure 11 – Deactivated access policies..... 33

98

Tables

99 Table 1 – Related profiles 12
 100 Table 2 – CIM_PolicyActivationService.ActivatePolicy() method: Parameters 21
 101 Table 3 – CIM_PolicyActivationService.ActivatePolicy() method: Return codes..... 21
 102 Table 4 – CIM_PolicyActivationService.DeactivatePolicy() method: Parameters 22
 103 Table 5 – CIM_PolicyActivationService.DeactivatePolicy() method: Return codes 22
 104 Table 6 – Operations: CIM_HostedService 22
 105 Table 7 – Operations: CIM_AssociatedPolicyActivationService 23
 106 Table 8 – Operations: CIM_ElementSettingData 23
 107 Table 9 – Operations: CIM_ElementCapabilities 23
 108 Table 10 – Operations: CIM_PolicySetComponent 24
 109 Table 11 – Operations: CIM_ReusablePolicy 24
 110 Table 12 – Operations: CIM_ServiceServiceDependency 24
 111 Table 13 – Operations: CIM_HostedAccessPoint 25
 112 Table 14 – Operations: CIM_ServiceAffectsElement 25
 113 Table 15 – Operations: CIM_ServiceAffectsElement 26
 114 Table 16 – CIM Elements: Integrated Access Control Policy Management Profile..... 34
 115 Table 17 – Class: CIM_HostedService 35
 116 Table 18 – Class: CIM_AssociatedPolicyActivationService 35
 117 Table 19 – Class: CIM_ElementSettingData 35
 118 Table 20 – Class: CIM_ElementCapabilities..... 36
 119 Table 21 – Class: CIM_PolicySetComponent..... 36
 120 Table 22 – Class: CIM_ReusablePolicyContainer 36
 121 Table 23 – Class: CIM_ReusablePolicy..... 37
 122 Table 24 – Class: CIM_ServiceServiceDependency 37
 123 Table 25 – Class: CIM_ServiceAffectsElement 37
 124 Table 26 – Class: CIM_AccessControlService 38
 125 Table 27 – Class: CIM_RegisteredProfile 38
 126 Table 28 – Class: CIM_AccessControlServiceSettingData 38
 127 Table 29 – Class: CIM_PolicyActivationService 39
 128 Table 30 – Class: CIM_PolicyTransferServiceAccessPoint 39
 129 Table 31 – Class: CIM_AccessControlPolicyGroup 39
 130 Table 32 – Class: CIM_AccessControlPolicy..... 40

131 Table 33 – Class: CIM_FileSystemCapabilities 40
132 Table 34 – Class: CIM_RelationalDatabaseCapabilities 40
133 Table 35 – Class: CIM_DatabaseContainsTable..... 41
134

135

Foreword

136 The *Integrated Access Control Policy Management Profile* (DSP1106) was prepared by the Policy
137 Working Group of the DMTF.

138 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
139 management and interoperability. For information about the DMTF, see <http://www.dmf.org>.

140 Acknowledgments

141 The DMTF acknowledges the following individuals for their contributions to this document:

- 142 • Mark Carlson – Sun
- 143 • Jorge Lobo – IBM
- 144 • Fumio Machida – NEC
- 145 • Masayuki Nakae – NEC
- 146 • Ryuichi Ogawa – NEC

147

148

Introduction

149 This document defines the classes used to compose and distribute common access control policies for
150 different access control components in managed systems. The information in this specification is intended
151 to be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties,
152 methods, and values that are mandatory to be instantiated and manipulated to represent and manage
153 users and groups that are modeled using the DMTF Common Information Model (CIM) core and
154 extended model definitions.

155 The target audience for this specification is implementers who are writing CIM based providers or
156 consumers of management interfaces representing the component described in this document.

157 Document conventions

158 Typographical conventions

159 The following typographical conventions are used in this document:

- 160 • Document titles are marked in *italics*.
- 161 • Important terms that are used for the first time are marked in *italics*.
- 162 • ABNF rules are in `monospaced font`.

163 ABNF usage conventions

164 Format definitions in this document are specified using ABNF (see [RFC5234](#)), with the following
165 deviations:

- 166 • Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the
167 definition in [RFC5234](#) that interprets literal strings as case-insensitive US-ASCII characters.

168

169 Integrated Access Control Policy Management Profile

170 1 Scope

171 The *Integrated Access Control Policy Management Profile* provides the ability for system administrators to
172 compose and distribute common access control policies for different access control components in
173 managed systems. The profile includes the models for target resources and resource-related capabilities
174 to compose the common access policies. A capability to control activation status of distributed policies is
175 also defined. This profile does not provide authentication or authorization in managed systems.

176 2 Normative references

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

180 DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
181 http://www.dmtf.org/sites/default/files/standards/documents/DSP0004_2.6.pdf

182 DMTF DSP0200, *CIM Operations over HTTP 1.3*,
183 http://www.dmtf.org/sites/default/files/standards/documents/DSP0200_1.3.pdf

184 DMTF DSP0231, *CIM Simplified Policy Language (CIM-SPL) 1.0*,
185 http://www.dmtf.org/sites/default/files/standards/documents/DSP0231_1.0.pdf

186 DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
187 http://www.dmtf.org/sites/default/files/standards/documents/DSP1001_1.0.pdf

188 DMTF DSP1033, *Profile Registration Profile 1.0*,
189 http://www.dmtf.org/sites/default/files/standards/documents/DSP1036_1.0.pdf

190 DMTF DSP1034, *Simple Identity Management Profile 1.0*,
191 http://www.dmtf.org/sites/default/files/standards/documents/DSP1034_1.0.pdf

192 DMTF DSP1039, *Role Based Authorization Profile 1.0*,
193 http://www.dmtf.org/sites/default/files/standards/documents/DSP1039_1.0.pdf

194 DMTF DSP1042, *System Virtualization Profile 1.0*,
195 http://www.dmtf.org/sites/default/files/standards/documents/DSP1042_1.0.pdf

196 DMTF DSP1057, *Virtual System Profile 1.0*,
197 http://www.dmtf.org/sites/default/files/standards/documents/DSP1057_1.0.pdf

198 IETF RFC5234, *Augmented BNF for Syntax Specifications: ABNF*
199 <http://tools.ietf.org/html/rfc5234>

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

202 **3 Terms and definitions**

203 In this document, some terms have a specific meaning beyond the normal English meaning. Those terms
204 are defined in this clause.

205 The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"),
206 "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described
207 in [ISO/IEC Directives, Part 2](#), Annex H. The terms in parenthesis are alternatives for the preceding term,
208 for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that
209 [ISO/IEC Directives, Part 2](#), Annex H specifies additional alternatives. Occurrences of such additional
210 alternatives shall be interpreted in their normal English meaning.

211 The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as
212 described in [ISO/IEC Directives, Part 2](#), Clause 5.

213 The terms "normative" and "informative" in this document are to be interpreted as described in [ISO/IEC](#)
214 [Directives, Part 2](#), Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do
215 not contain normative content. Notes and examples are always informative elements.

216 The terms defined in [DSP0004](#), [DSP0200](#), and [DSP1001](#) apply to this document. The following additional
217 terms are used in this document.

218 **3.1**

219 **access control policy activation service**

220 a service component on the managed system for enabling and disabling access control policies on target
221 access control services

222 **3.2**

223 **access control policy composition**

224 specifies access control policies by assembling principals, resources and actions on managed systems

225 **3.3**

226 **access control policy distribution**

227 transfers access control policies to managed systems through access policy transfer services, and
228 activate the transferred policies through access activation services

229 **3.4**

230 **access control policy group**

231 a collection of the access policies of a particular access control service

232 **3.5**

233 **access control policy transfer service access point**

234 an ingress point to access a particular access control policy transfer service that is used to transfer
235 arbitrary policy descriptions from a remote host to a managed system

236 **3.6**

237 **access control service**

238 a service component that enforces specified access control policies in order to protected resources on a
239 managed system

240 **3.7**

241 **action type**

242 a class of operations to resources controlled by a certain access control service (for example, read, write,
243 execute)

244 **3.8**245 **activation status**

246 a status of access policy activation process

247 **3.9**248 **principal type**

249 a class of principals supported by a certain access control service (for example, users, accounts, groups)

250 **3.10**251 **resource type**252 a class of target objects supported by a certain access control service (for example, files, virtual
253 machines, data bases)254 **4 Symbols and abbreviated terms**

255 The following abbreviations are used in this document.

256 **4.1**257 **IAM**

258 Integrated Access Control Manager

259 **4.2**260 **URI**

261 universal resource identifier

262 **4.3**263 **XACML**

264 Extensible Access Control Markup Language

265 **5 Synopsis**266 **Profile Name:** Integrated Access Control Policy Management267 **Version:** 1.0.0268 **Organization:** DMTF269 **CIM schema version:** 2.19270 **Central Class:** CIM_AccessControlService271 **Scoping Class:** CIM_ComputerSystem272 The *Integrated Access Control Policy Management Profile* provides the ability to compose and distribute
273 common access control policies for different access control components in managed systems.274 The Central Class of the *Integrated Access Control Policy Management Profile* shall be
275 CIM_AccessControlService. The Central Instance shall be an instance of CIM_AccessControlService.
276 The Scoping Class shall be CIM_ComputerSystem. The Scoping Instance shall be the instance of
277 CIM_ComputerSystem associated to the Central Instance through the CIM_HostedService association.278 Table 1 lists the profiles related to the *Integrated Access Control Policy Management Profile*.

279

Table 1 – Related profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	
Role Based Authorization	DMTF	1.0	Optional	
Simple Identity Management	DMTF	1.0	Optional	

280 6 Description

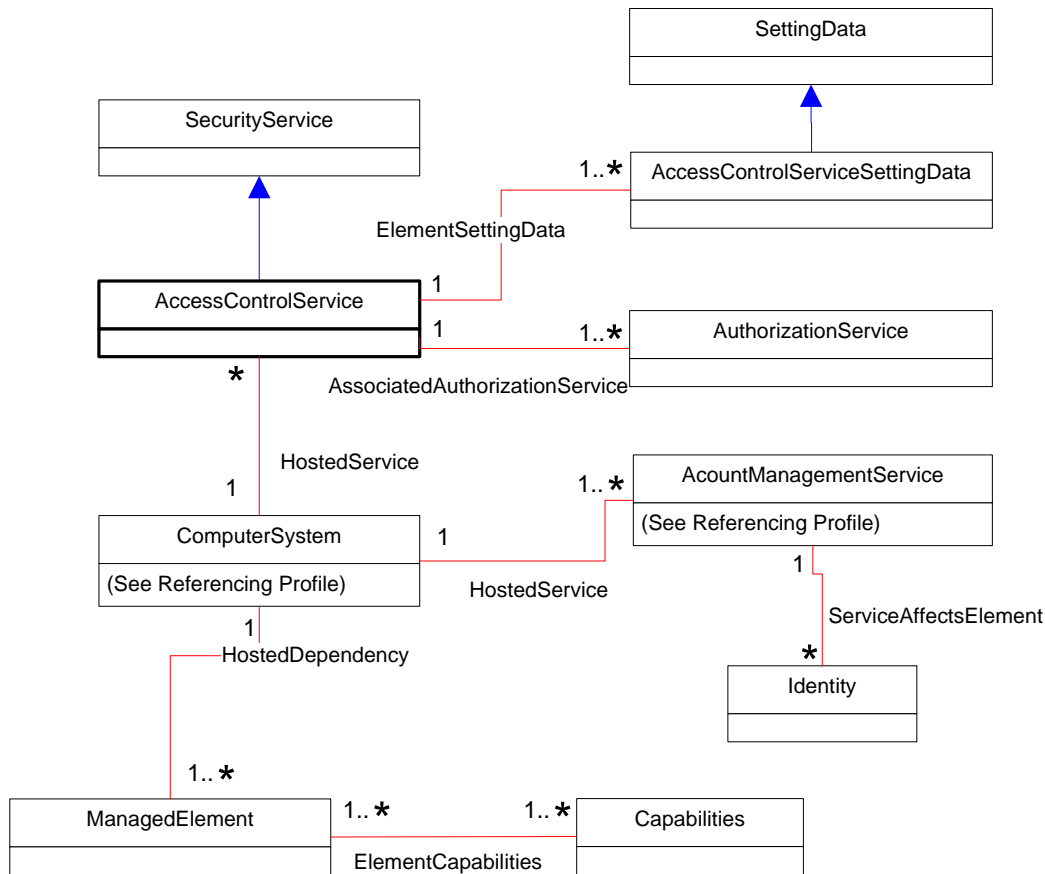
281 The *Integrated Access Control Policy Management Profile* provides the ability to compose and distribute
 282 common access control policies for different access control components in managed systems. This profile
 283 is separated into two parts: access control policy composition and distribution.

284 6.1 Access control policy composition

285 6.1.1 General

286 Figure 1 represents the class schema for the access control policy composition part of the *Integrated*
 287 *Access Control Management Profile*. For simplicity, the prefix CIM_ has been removed from the names of
 288 the classes.

289 This subclause describes models that are used to compose access control policies.



290

291

Figure 1 – Access control policy composition: Class diagram

292 **6.1.2 Access control service**

293 An instance of CIM_AccessControlService represents an access control service that controls access to
 294 the target resources by given access control policies. The CIM_AccessControlService instance extends
 295 the CIM_SecurityService instance and is associated with the CIM_ComputerSystem instance through the
 296 CIM_HostedService association. The CIM_AccessControlService instance is also associated with the
 297 CIM_AuthorizationService instance through the CIM_AssociatedAuthorizationService association. The
 298 instances of CIM_AccessControlService shall be mapped to either a software module or hardware device
 299 to perform access control.

300 **6.1.3 Principal type and resource type**

301 Principal type and resource type supported by the access control services are represented by the
 302 instances of CIM_AccessControlServiceSettingData that are associated with the instances of
 303 CIM_AccessControlService through the CIM_ElementSettingData association. The PrincipalType
 304 property of the CIM_AccessControlServiceSettingData instance specifies the principal types (for example,
 305 users, accounts, and groups). The ResourceType property of the CIM_AccessControlServiceSettingData
 306 instance specifies the types of the managed elements to be accessed (for example, file system, virtual
 307 machine, RDB Table, database, and table column).

308 Referring to the principal and resource types supported by the access control services through the
 309 instance of CIM_AccessControlServiceSettingData, the system administrator identifies the instances of

310 principals (CIM_Identity) and managed elements (CIM_ManagedElement) for composing access control
311 policies.

312 **6.1.4 Resource-related capabilities**

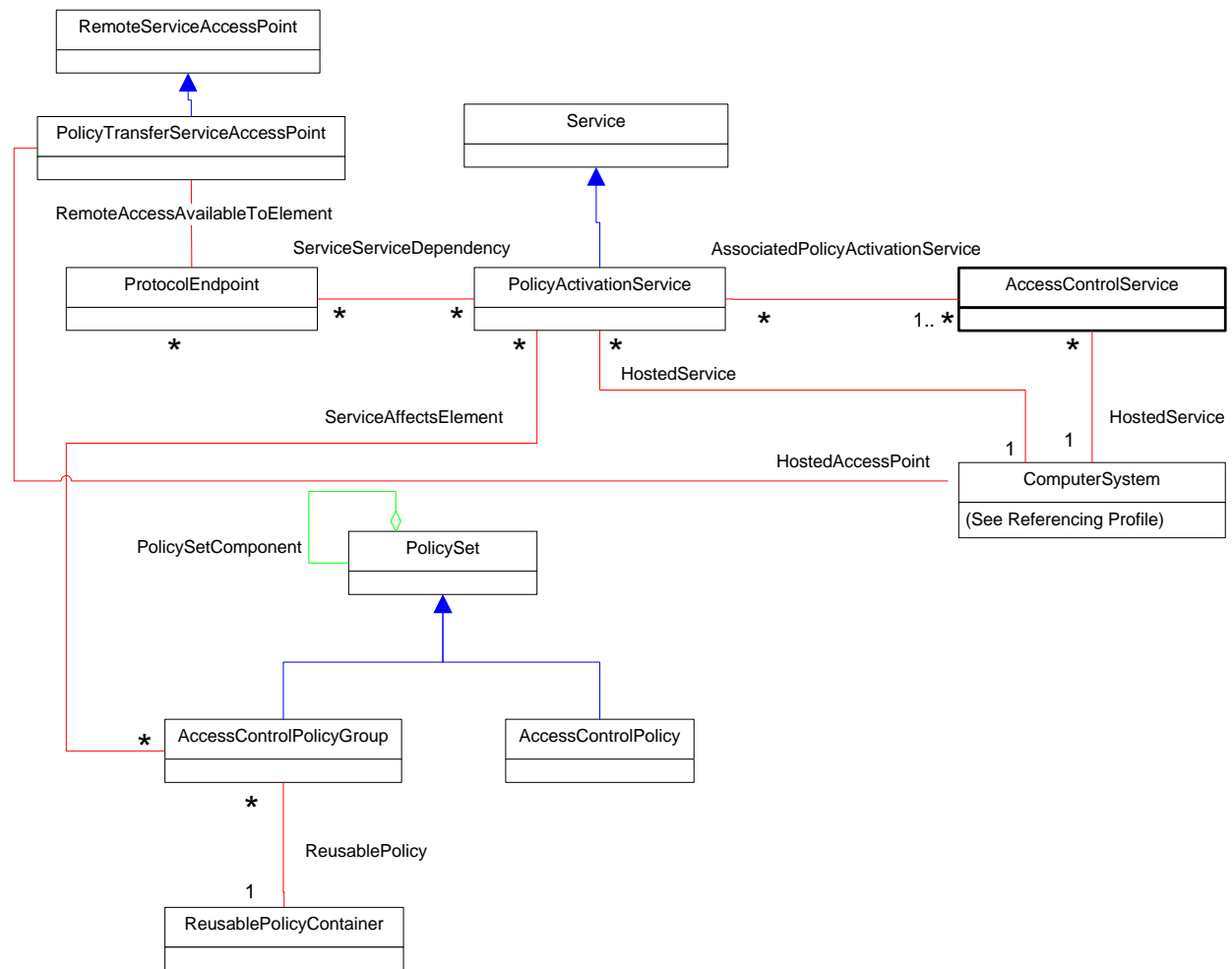
313 Action types used in the access control policies correspond to the operations to target resources. The set
314 of operations to target resources are represented by the properties of CIM_Capabilities subclasses that
315 are associated with the resource types. To compose the access control policies for an access control
316 service, the system administrator identifies the resource type supported by the access control service
317 through the instance of CIM_AccessControlServiceSettingData. According to the identified resource type,
318 the system administrator finds the instance of the subclass of CIM_Capabilities that are related to the
319 identified resource type. The properties of the referred CIM_Capabilities subclass are used as action
320 types for access control policies.

321 **6.2 Access control policy distribution**

322 **6.2.1 General**

323 Figure 2 represents the class schema for the access control policy distribution part of the *Integrated*
324 *Access Control Management Profile*. For simplicity, the prefix CIM_ has been removed from the names of
325 the classes.

326 This clause describes models that are used for the policy distribution and activation on the target
327 managed system.



328

329

Figure 2 – Access control policy distribution: Class diagram

330 6.2.2 Policy transfer service access point

331 The CIM_PolicyTransferServiceAccessPoint class represents the ingress points of data transfer services
 332 (FTP, HTTP, etc.) for distributing access control policy descriptions from a remote system to the managed
 333 system. The PolicyTransferURIs property of the CIM_PolicyTransferServiceAccessPoint instance
 334 indicates the destination URIs.

335 6.2.3 Access control policy

336 The distributed access policies on the managed systems are represented by the instances of
 337 CIM_AccessControlPolicy that extend the CIM_PolicySet instance. Each CIM_AccessControlPolicy
 338 instance shall have the PolicyID property that is a unique identifier in an access policy group. The
 339 Enabled property inherited from CIM_PolicySet represents the activation status of an access policy for a
 340 certain access control service.

341 A set of access control policies is represented by an instance of CIM_AccessControlPolicyGroup that
 342 extends the CIM_PolicySet class. An instance of CIM_AccessControlPolicyGroup shall be associated
 343 with instances of CIM_AccessControlPolicy through the CIM_PolicySetComponent association.

344 **6.2.4 Policy activation service**

345 The CIM_PolicyActivationService class represents the services that enable and disable the transferred
346 policies on a target access control service. The CIM_PolicyActivationService class, through extrinsic
347 methods, serves as the interface for applying and removing access control policies to the associated
348 access control services. After the access policy application, the Enable property of the
349 CIM_AccessControlPolicy instance is changed to TRUE. After the access control policy removal, the
350 Enable property of the CIM_AccessControlPolicy instance is changed to FALSE.

351 The instances of CIM_PolicyActivationService are associated with the instances of
352 CIM_AccessControlService through the CIM_AssociatedPolicyActivationService association.

353 A CIM_PolicyActivationService instance may have an association to a CIM_ProtocolEndpoint instance
354 that is also associated with a CIM_PolicyActivationTransferAccessPoint instance. These associations
355 clarify the correspondence between policy activation services and policy transfer services.

356 **7 Implementation**

357 This clause details the requirements related to the arrangement of instances and their properties for
358 implementations of this profile.

359 **7.1 Access control service**

360 **7.1.1 General**

361 An access control service on the managed system shall be represented by an instance of the
362 CIM_AccessControlService class. An instance of CIM_AccessControlService shall be associated to only
363 one instance of a CIM_ComputerSystem through a CIM_HostedService association. This instance of
364 CIM_ComputerSystem shall be the Scoping Instance.

365 An instance of CIM_AccessControlService shall be associated with more than one instance of the
366 CIM_AccessControlServiceSettingData class that represents the supported access control types of the
367 installed access control service. A supported principal type is specified in the
368 CIM_AccessControlServiceSettingData.PrincipalType property. For each supported principal type,
369 corresponding supported resource types are specified in the
370 CIM_AccessControlServiceSettingData.ResourceType property.

371 An instance of CIM_AccessControlService may be associated with an instance of
372 CIM_AuthorizationService through an instance of the CIM_SoftwareElementServiceImplementation
373 association. The access control service may be managed locally through the methods of the
374 CIM_AuthorizationService class.

375 **7.1.2 Version**

376 The version of the implementation of the access control service shall be specified in the Version property
377 that is one of the key properties of the CIM_AccessControlService class. Two different versions of an
378 access control service may be installed on the same computer system. In such a case, the value of the
379 CIM_AccessControlService.Version property is used to identify the specific version of the access control
380 service.

381 **7.1.3 Implementation type (optional)**

382 The implementation type of the access control service may be specified in the ImplementationType
383 property of an instance of CIM_AccessControlService. When the
384 CIM_AccessControlService.ImplementationType property has the value 1 (OS module), the access
385 control service is implemented as an embedded software module in the operating system.

386 When the CIM_AccessControlService.ImplementationType property has the value 2 (Application), the
387 access control service is implemented as an application running on the operating system.

388 When the CIM_AccessControlService.ImplementationType property has the value 3 (Hardware), the
389 access control service is implemented as a device of the computer system.

390 7.2 Principals

391 All principals supported by the specific access control service may be represented by instances of
392 CIM_Identity. The type of principal is specified in the PrincipalType property of the
393 CIM_AccessControlServiceSettingData instance. According to the specified principal type (for example,
394 accounts and groups), the system administrator finds the instances of CIM_Identity to compose the
395 access control policies. Each instance of CIM_Identity represents an account, user, or group. The
396 identities of principals may be managed by the instance of CIM_AccountManagementService as
397 described in the [Simple Identity Management Profile](#).

398 7.3 Resources and resource-related capabilities

399 7.3.1 General

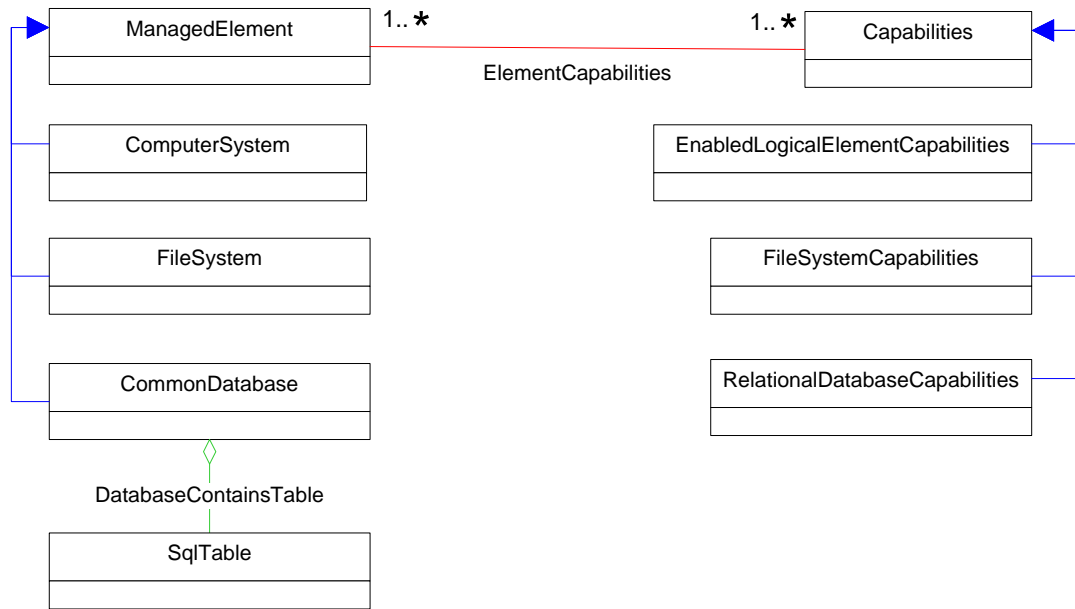
400 All resources that are managed by the specific access control service shall be represented by instances
401 of CIM_ManagedElement subclasses. The resource type is specified in the ResourceType property of the
402 CIM_AccessControlServiceSettingData instance. The system administrator finds the instances of
403 CIM_ManagedElement corresponding to the specified resource type (for example, files and directories, or
404 virtual machines) to compose the access control policies.

405 The set of operations that are managed by the IAM as the action types shall be represented by the value
406 map property of the CIM_Capabilities subclasses. The instance of CIM_Capabilities is associated with the
407 CIM_ManagedElement instances representing the resources to be accessed or a resource management
408 service through the instance of the CIM_ElementCapabilities associations. For example, when access to
409 files and directories is allowed through a certain file system (for example, ext3 in Linux), the set of
410 operations for files and directories is represented by the value map property of
411 CIM_FileSystemCapabilities.

412 Additional CIM_Capabilities instances may be associated with the above CIM_ManagedElement instance.

413 This profile conditionally supports the following resources and their capabilities, as shown in Figure 3:

- 414 • Virtual machine: CIM_ComputerSystem and CIM_EnabledLogicalElementCapabilities
- 415 • File system: CIM_FileSystem and CIM_FileSystemCapabilities
- 416 • Database: CIM_CommonDatabase and CIM_RelationalDatabaseCapabilities



417

418

Figure 3 – Supported resources and capabilities

419 **7.3.2 Virtual machines (conditional)**

420 If an access control service controls actions to virtual machines, the
 421 CIM_AccessControlServiceSettingData.ResourceType property is set to 3 (Virtual Machine). All target
 422 machines are represented by instances of CIM_ComputerSystem that are associated with an instance of
 423 CIM_VirtualSystemManagementService through the CIM_ServiceAffectsElement association.

424 Action types for virtual machines are modeled by the CIM_EnabledLogicalElementCapabilities instance
 425 that is associated to an instance of CIM_ComputerSystem through the CIM_ElementCapabilities
 426 association. The CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property array
 427 contains the values defined in [DSP1057](#), including: 2 (Enabled), 3 (Disabled), and 4 (Shut Down), 6
 428 (Offline), 9 (Quiesce), 10 (Reboot), and 11 (Reset).

429 **7.3.3 Files and directories (conditional)**

430 If an access control service controls actions to files and directories, the
 431 CIM_AccessControlServiceSettingData.ResourceType property is set to 2 (File System). All target files
 432 and directories are represented by instances of CIM_LogicalFile that are associated with an instance of
 433 CIM_FileSystem through the instance of CIM_FileStorage aggregation.

434 Action types for files and directories are modeled by the CIM_FileSystemCapabilities instance that is
 435 associated to an instance of CIM_FileSystem through the CIM_ElementCapabilities association. The
 436 CIM_FileSystemCapabilities.SupportedOperations property array contains the following values: 1 (Read),
 437 2 (Write), and 3 (Execute).

438 **7.3.4 Database tables (conditional)**

439 If an access control service controls actions to tables of relational databases, the
 440 CIM_AccessControlServiceSettingData.ResourceType property is set to 4 (RDB Table). All target tables
 441 are represented by instances of CIM_SqlTable that are associated with an instance of
 442 CIM_CommonDatabase through an instance of CIM_DatabaseContainsTable. The instance of

443 CIM_CommonDatabase is associated with an instance of CIM_DatabaseService through a
444 CIM_ServiceAffectsElement association.

445 Similarly, when databases themselves are controlled targets,
446 CIM_AccessControlServiceSettingData.ResourceType shall be set to 5 (Database), and target databases
447 are represented by CIM_CommonDatabase instances.

448 Supported operations for tables and databases include SQL92 standard commands (for example, alter,
449 create, drop, grant, delete, insert, index, lock, references, select, and update). The SQL operations are
450 modeled by the CIM_RelationalDatabaseCapabilities instance that is associated with the
451 CIM_DatabaseService instance through the CIM_ElementCapabilities association.

452 In CIM_RelationalDatabaseCapabilities, the supported operations are separately modeled according to a
453 general database structure, which means that supported operations for tables are represented by the
454 SupportedTableOperations property, and supported operations for databases are represented by the
455 SupportedDBOperations property. For example, some of the values in the
456 CIM_RelationalDatabaseCapabilities.SupportedTableOperations property are 1 (Alter), 2 (Grant), and 3
457 (Insert), which show that the listed SQL commands can be executed to any table.

458 **7.4 Policy transfer service access point**

459 **7.4.1 General**

460 A CIM_PolicyTransferServiceAccessPoint instance represents an end point of a policy transfer service
461 that is used for policy distribution on a managed system. At least one instance of the
462 CIM_PolicyTransferServiceAccessPoint class shall exist on a managed system.

463 An instance of CIM_PolicyTransferServiceAccessPoint shall be associated with a CIM_ComputerSystem
464 instance through the CIM_HostedAccessPoint dependency. Also, the instance shall be associated with a
465 CIM_ProtocolEndpoint instance (that is, a service component of policy transfer) through
466 CIM_RemoteAccessAvailableToElement, which is associated with a CIM_PolicyActivationService
467 instance through a CIM_ServiceServiceDependency association.

468 **7.4.2 Policy transfer URI**

469 The PolicyTransferURIs property of the CIM_PolicyTransferServiceAccessPoint class shall hold at least
470 one universal resource identifier for transferring access policies to the managed systems. The property
471 shall be initialized with an array of URI strings, each of which represents a transfer protocol (for example,
472 FTP, SCP, and NFS) and a destination of policy transfer. These values shall be immutable as long as the
473 CIM_PolicyTransferServiceAccessPoint instance exists.

474 **7.5 Access control policy**

475 **7.5.1 General**

476 An instance of CIM_AccessControlPolicy represents an access control policy description (for example, a
477 file and an XACML's policy element) and its metadata (for example, the policy identifier and activation
478 status). With the creation and deletion of an access policy, a corresponding CIM_AccessControlPolicy
479 instance shall be instantiated and deleted. Each CIM_AccessControlPolicy instance shall have at least
480 two properties of PolicyID and Enabled for representing its policy identifier and activation status,
481 respectively. For more details about these properties, see 7.5.2 and 7.5.3.

482 An instance of CIM_AccessControlPolicyGroup represents a collection of access policies for a certain
483 access control service, and it shall be associated with CIM_AccessControlPolicy instances through a
484 CIM_PolicySetComponent association.

485 For each CIM_PolicyActivationService instance, at least one CIM_AccessControlPolicyGroup instance
486 shall exist and be associated with the corresponding CIM_PolicyActivationService instance through the
487 CIM_ServiceAffectsElement association.

488 **7.5.2 Policy identifier**

489 The PolicyID property is a key property of CIM_AccessControlPolicy; it holds an identifier of an access
490 policy. The identifier shall be an immutable octet string.

491 A value of the PolicyID property shall be unique in the scope of CIM_AccessControlPolicyGroup. In other
492 words, CIM_AccessControlPolicy instances associated with an identical CIM_AccessControlPolicyGroup
493 instance shall be distinguishable by the PolicyID properties.

494 **7.5.3 Activation status**

495 The activation status of an access policy shall be reflected in the Enabled property of a
496 CIM_AccessControlPolicy instance. The Enabled property is an inherited property from the
497 CIM_PolicySet class; it shall be FALSE if the CIM_AccessControlPolicy instance is inactive, and it shall
498 be TRUE if the CIM_AccessControlPolicy instance is active.

499 Right after a policy is created or distributed onto the managed system, the corresponding
500 CIM_AccessControlPolicy.Enabled property shall be FALSE. When the
501 CIM_PolicyActivationService.ActivatePolicy() method successfully completes, the Enabled property shall
502 be changed to TRUE. Also, when the CIM_PolicyActivationService.DeactivatePolicy() method
503 successfully completes, the Enabled property shall be changed to FALSE.

504 **7.6 Policy activation service**

505 CIM_PolicyActivationService represents a service to enable and disable the distributed access policy on
506 associated access control services. For each CIM_AccessControlService instance, at least one
507 CIM_PolicyActivationService shall exist. The instances are associated with one another through the
508 CIM_AssociatedPolicyActivationService association.

509 CIM_PolicyActivationService is also associated with the CIM_AccessControlPolicyGroup instance
510 through the CIM_ServiceAffectsElement association.

511 CIM_PolicyActivationService shall support the ActivatePolicy() and DeactivatePolicy() methods in order
512 to execute policy activation and deactivation, respectively. The ActivatePolicy() method applies access
513 control policies to the target access control service, and the Enabled property of target
514 CIM_AccessControlPolicy instances are changed to true. The DeactivatePolicy() method removes
515 access control policies from the target access control service, and the Enabled properties of target
516 CIM_AccessControlPolicy instances are changed to FALSE.

517 **8 Methods**

518 This clause details the requirements for supporting intrinsic operations and extrinsic methods for the CIM
519 elements defined by this profile.

520 **8.1 Extrinsic methods**

521 The CIM_PolicyActivationService class shall support the two extrinsic methods: the ActivatePolicy()
522 method and the DeactivatePolicy() method.

523 **8.1.1 CIM_PolicyActivationService.ActivatePolicy()**

524 The ActivatePolicy() method is used to activate the collection of access policies specified with the
525 TargetPolicies parameter (see Table 2).

526 Upon the successful execution of the ActivatePolicy() method, the following actions occur:

- 527 • When the PolicyID parameter is Null, the instance of CIM_PolicyActivationService does not activate
528 any access policies.
- 529 • When the TargetPolicies parameter is not Null, the ActivatePolicy() method enables the
530 CIM_AccessControlPolicy instances specified in the TargetPolicies parameter. More specifically, the
531 ActivatePolicy() method sets the access policy rules for the access control service corresponding to
532 the instance of CIM_AccessControlService associated with the instance of
533 CIM_PolicyActivationService through the CIM_AssociatedPolicyActivationService association. As a
534 result, the method changes the Enabled property of each CIM_AccessControlPolicy instance to
535 TRUE.

536 The ActivatePolicy() method shall return the value 1 (Failed) in the following cases:

- 537 • The TargetPolicies parameter includes an identifier to reference no CIM_AccessControlPolicy
538 instance.
- 539 • Applying the access policies to the target access control service did not execute successfully.

540 The ActivatePolicy() method's parameters are specified in Table 2, and its return codes are specified in
541 Table 3.

542 **Table 2 – CIM_PolicyActivationService.ActivatePolicy() method: Parameters**

Qualifiers	Name	Type	Description
IN, REQ	TargetPolicies	CIM_AccessControlPolicy[] REF	Array of CIM_AccessControlPolicy instances to be activated

543 **Table 3 – CIM_PolicyActivationService.ActivatePolicy() method: Return codes**

Value	Description
0	Activation completed successfully.
1	Failed

544 **8.1.2 CIM_PolicyActivationService.DeactivatePolicy()**

545 The DeactivatePolicy() method is used to deactivate the collection of access policies specified with the
546 TargetPolicies parameter (see Table 4).

547 Upon the successful execution of the DeactivatePolicy() method, the following actions occur:

- 548 • When the PolicyID parameter is Null, the instance of CIM_PolicyActivationService does not deactivate
549 any access policies.
- 550 • When the TargetPolicies parameter is not Null, the DeactivatePolicy() method disables the
551 CIM_AccessControlPolicy instances specified in the TargetPolicies parameter. More specifically, the
552 method revokes the access policy rules in the access control service corresponding to the instance
553 of CIM_AccessControlService associated with the CIM_PolicyActivationService through the
554 CIM_AssociatedPolicyActivationService association. As a result, the method changes the Enabled
555 property of each CIM_AccessControlPolicy instance to FALSE.

556 The DeactivatePolicy() method shall return the value 1 (Failed) in the following cases:

- 557 • The TargetPolicies parameter includes an identifier to reference no CIM_AccessControlPolicy
- 558 instance.
- 559 • Revoking the access policies for the target access control service did not execute successfully.

560 The DeactivatePolicy() method's parameters are specified in Table 4, and its return codes are specified

561 in Table 5.

562 **Table 4 – CIM_PolicyActivationService.DeactivatePolicy() method: Parameters**

Qualifiers	Name	Type	Description
IN, REQ	TargetPolicies	CIM_AccessControlPolicy[] REF	Array of CIM_AccessControlPolicy instances to be deactivated

563 **Table 5 – CIM_PolicyActivationService.DeactivatePolicy() method: Return codes**

Value	Description
0	Deactivation completed successfully.
1	Failed

564 **8.2 Profile conventions for operations**

565 **8.2.1 General**

566 For each profile class (including associations), the implementation requirements for operations, including

567 those in the following default list, are specified in class-specific subclauses of this clause.

568 The default list of operations is as follows:

- 569 • GetInstance
- 570 • Associators
- 571 • AssociatorNames
- 572 • References
- 573 • ReferenceNames
- 574 • EnumerateInstances
- 575 • EnumerateInstanceNames

576 **8.2.2 CIM_HostedService**

577 Table 6 lists implementation requirements for operations. If implemented, these operations shall be

578 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 6, all operations in

579 the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

581 **Table 6 – Operations: CIM_HostedService**

Operation	Requirement	Messages
Associators	Unspecified	None

AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

582 **8.2.3 CIM_AssociatedPolicyActivationService**

583 Table 7 lists implementation requirements for operations. If implemented, these operations shall be
 584 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 7, all operations in
 585 the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

587 **Table 7 – Operations: CIM_AssociatedPolicyActivationService**

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

588 **8.2.4 CIM_ElementSettingData**

589 Table 8 lists implementation requirements for operations. If implemented, these operations shall be
 590 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 8, all operations in
 591 the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

593 **Table 8 – Operations: CIM_ElementSettingData**

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

594 **8.2.5 CIM_ElementCapabilities**

595 Table 9 lists implementation requirements for operations. If implemented, these operations shall be
 596 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 9, all operations in
 597 the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

599 **Table 9 – Operations: CIM_ElementCapabilities**

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None

References	Unspecified	None
ReferenceNames	Unspecified	None

600 **8.2.6 CIM_PolicySetComponent**

601 Table 10 lists implementation requirements for operations. If implemented, these operations shall be
 602 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 10, all operations
 603 in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

605 **Table 10 – Operations: CIM_PolicySetComponent**

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

606 **8.2.7 CIM_ReusablePolicy**

607 Table 11 lists implementation requirements for operations. If implemented, these operations shall be
 608 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 11, all operations
 609 in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

611 **Table 11 – Operations: CIM_ReusablePolicy**

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

612 **8.2.8 CIM_ServiceServiceDependency**

613 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
 614 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations
 615 in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

617 **Table 12 – Operations: CIM_ServiceServiceDependency**

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

ReferenceNames	Unspecified	None
----------------	-------------	------

618 **8.2.9 CIM_HostedAccessPoint**

619 Table 13 lists implementation requirements for operations. If implemented, these operations shall be
 620 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 13, all operations
 621 in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

623 **Table 13 – Operations: CIM_HostedAccessPoint**

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

624 **8.2.10 CIM_ServiceAffectsElement**

625 Table 14 lists implementation requirements for operations. If implemented, these operations shall be
 626 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 14, all operations
 627 in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

629 **Table 14 – Operations: CIM_ServiceAffectsElement**

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

630 **8.2.11 CIM_AccessControlService**

631 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

633 **8.2.12 CIM_AccessControlServiceSettingData**

634 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

636 **8.2.13 CIM_PolicyActivationService**

637 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

639 8.2.14 CIM_PolicyTransferServiceAccessPoint

640 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

642 8.2.15 CIM_AccessControlPolicyGroup

643 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

645 8.2.16 CIM_AccessControlPolicy

646 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

648 8.2.17 CIM_ProtocolEndpoint

649 All operations in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

651 8.2.18 CIM_RemoteAccessAvailableToElement

652 Table 15 lists implementation requirements for operations. If implemented, these operations shall be
 653 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 15, all operations
 654 in the default list in 8.2.1 shall be implemented as defined in [DSP0200](#).

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

656 **Table 15 – Operations: CIM_ServiceAffectsElement**

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

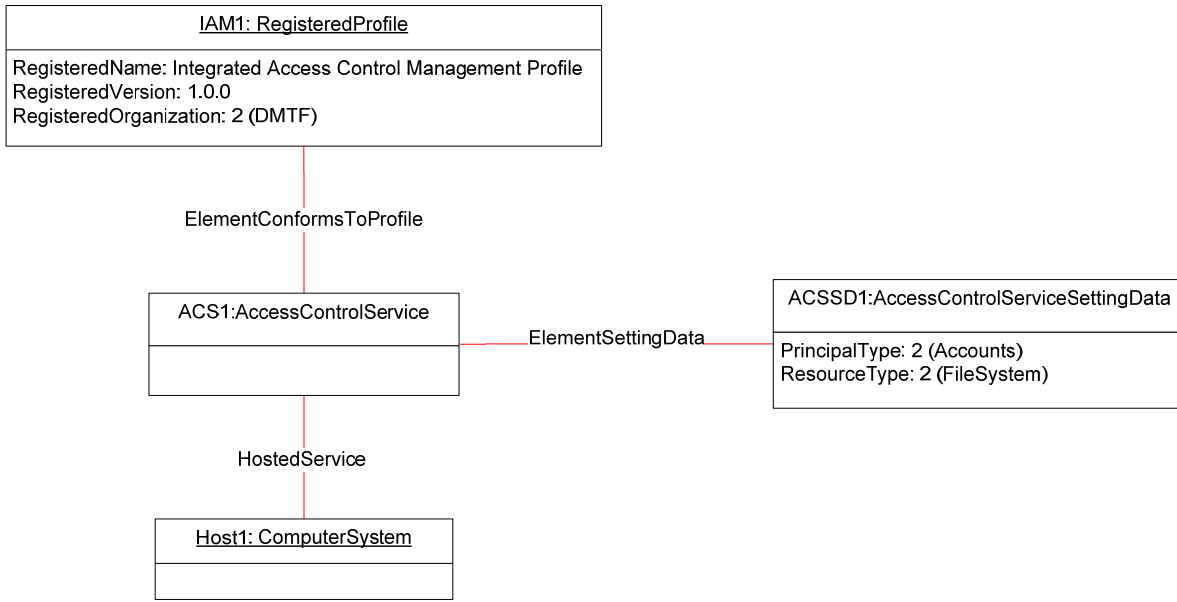
657 9 Use cases

658 This clause contains object diagrams and use cases for the *Integrated Access Control Policy*
 659 *Management Profile*.

660 The contents of this clause are for informative purposes only and do not constitute normative
 661 requirements for implementations of this specification.

662 9.1 Discover conformant access control service

663 An access control service supporting the *Integrated Access Control Management Profile* can be
 664 discovered through an instance of CIM_RegisteredProfile with the Central Class methodology or Scoping
 665 Class methodology. If the Central Class methodology is used, the instance of CIM_RegisteredProfile that
 666 represents the *Integrated Access Control Management Profile* is directly found with the target access
 667 control service through the CIM_ElementConformsToProfile association.



668

669

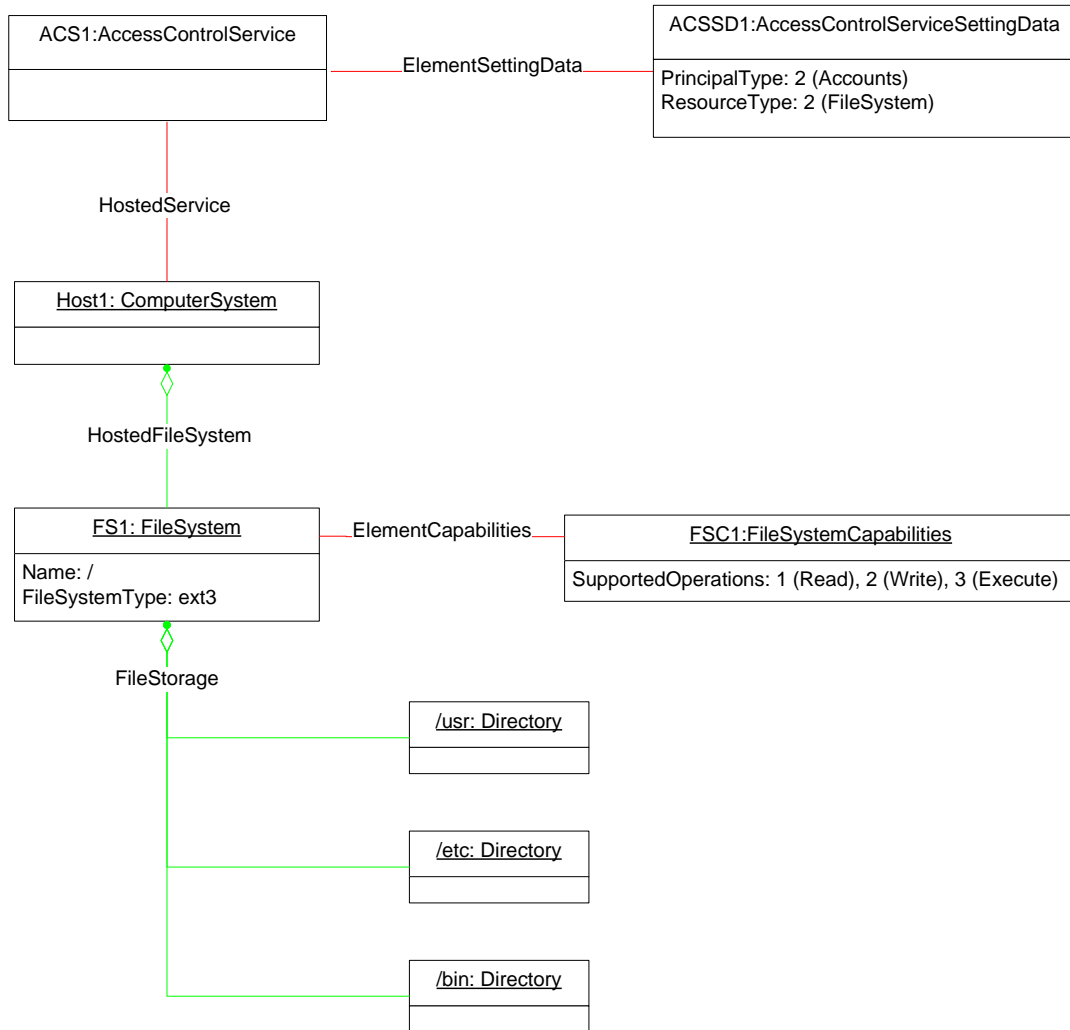
Figure 4 – IAM registered profile and access control service

670 Figure 4 illustrates a system hosting an IAM-conformant access control service for a file system. The host
 671 system is represented by an instance of the CIM_ComputerSystem class named Host1. The host system
 672 hosts an access control service represented by an instance of the CIM_AccessControlService class
 673 labeled ACS1. ACS1 is associated with an instance of CIM_AccessControlServiceSettingData labeled
 674 ACSSD1 that represents the principal type and the resource type supported by ACS1. ACS1 is
 675 discovered as an IAM-conformant implementation through the CIM_ElementConformsToProfile
 676 association from the instance of CIM_RegisteredProfile named IAM1. IAM1 contains the profile name and
 677 version information.

678 **9.2 Determine the principal type and the resource type**

679 The principal type and the resource type supported by ACS1 are represented in the properties of
 680 ACSSD1. The value of the PrincipalType property of ACSSD1 is set to 2 (Accounts), indicating that ACS1
 681 controls resource access for each account on the system. The value of the ResourceType property of
 682 RMT1 is set to 2 (FileSystem), indicating that RM1 controls access to files and directories on the system.

683 **9.3 Determine the resource related capabilities**



684

685 **Figure 5 – Hosted file system and related capabilities**

686 Figure 5 depicts the hosted file systems and its related capabilities information. The specific file system
 687 implemented on Host1 is represented by an instance of CIM_FileSystem named FS1. FS1 is associated
 688 with Host1 by the CIM_HostedFileSystem instance. All files and directories managed by FS1 are
 689 represented by the instances of CIM_LogicalFile and CIM_Directory. These instances are aggregated
 690 into FS1 through the CIM_FileStorage association.

691 The set of actions to the files and directories managed by FS1 are represented in an instance of
 692 CIM_FileSystemCapabilities labeled FSC1 that is associated with FS1 through an instance of
 693 CIM_ElementCapabilities. Supported operations of FS1 include all the values of the SupportedOperations
 694 property: 1 (Read), 2 (Write), and 3 (Execute). The set of actions may vary according to the
 695 implementation of the file system.

696 **9.3.1 Other resource types**

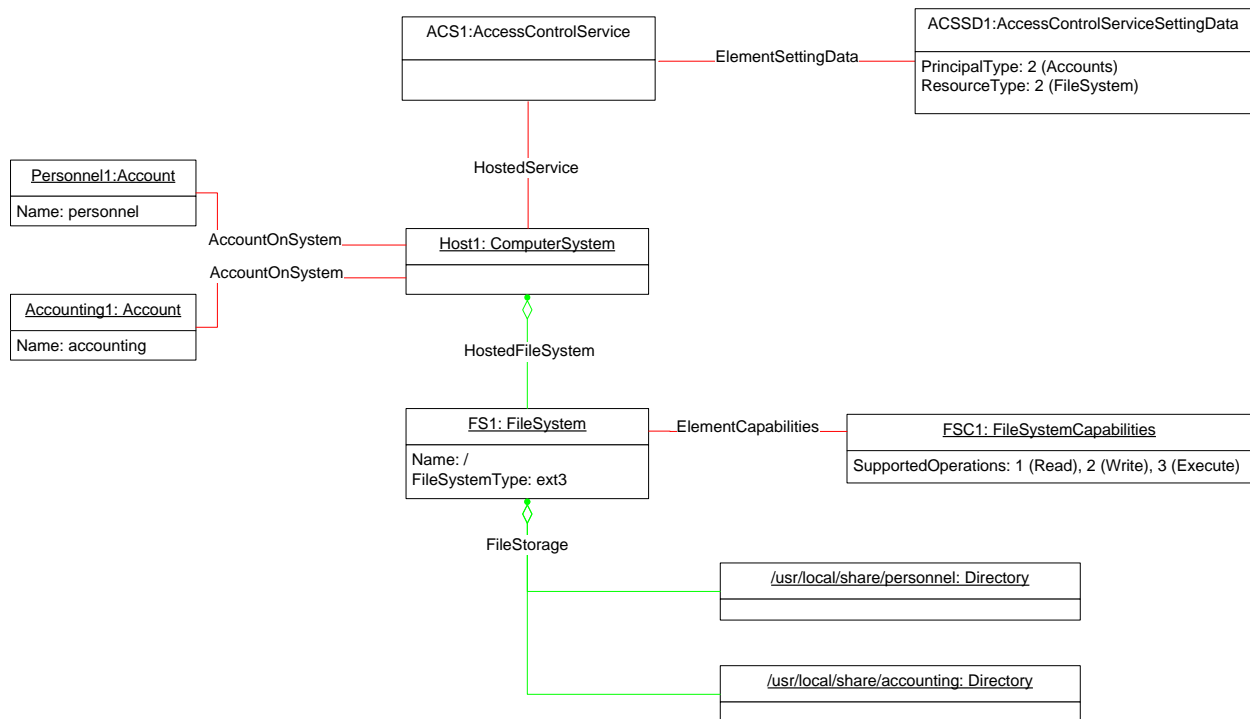
697 **9.3.1.1 Virtual machines**

698 For the different types of resources, the action types are represented by an instance of the capabilities
 699 class associated with the resources. If the value of the ResourceType property of the
 700 CIM_AccessControlServiceSettingData instance is "Virtual Machine", the set of actions for virtual
 701 machine resources is represented by an instance of CIM_EnabledLogicalElementCapabilities that is
 702 associated with an instance of CIM_ComputerSystem by an instance of the CIM_ElementCapabilities
 703 association.

704 **9.3.1.2 Relational databases**

705 If the ResourceType property is set to "RDB Table", the set of actions for RDB tables is represented by
 706 the CIM_RelationalDatabaseCapabilities instance that is associated with the instance of
 707 CIM_CommonDatabase by an instance of the CIM_ElementCapabilities association.

708 **9.4 Compose access control policies**



709

710 **Figure 6 – Accounts and resources for policy composition**

711 Figure 6 shows a system that has two local accounts and two directories that need to be protected from
 712 unauthorized access. FS1 is shared with two different divisions in a company: the personnel division and
 713 the accounting management division. The users in the personnel division use the "personnel" account on
 714 Host1 that is represented by an instance of CIM_Account named Personnel1. The users in the
 715 accounting management division use the "accounting" account that is represented by an instance of
 716 CIM_Account named Accounting1. Personnel1 and Accounting1 are associated with Host1 through
 717 instances of the CIM_AccountOnSystem association.

718 FS1 has exclusive directories for each division. The directory "/usr/local/share/personnel" is accessible
 719 only by users in the personnel division. The directory "/usr/local/share/accounting" is accessible only by
 720 users in the accounting division. These directories should be protected from access by outsiders. The
 721 access control service ACS1 performs access control in accordance with these access control policies.

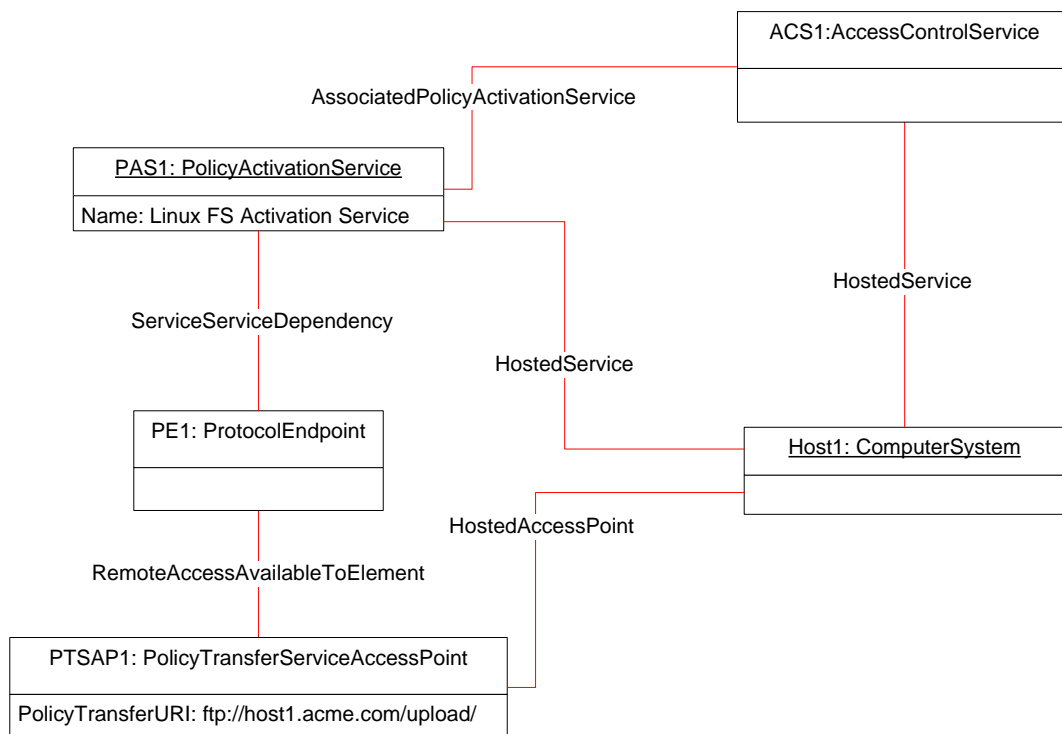
722 To compose access control policies for ACS1 in an integrated manner, the principals, resources and
 723 resource-related actions need to be determined as follows:

- 724 1) The type of principal supported by RM1 is determined by ACSSD1. Because the value of the
 725 PrincipalType property is "Accounts", the system administrator finds the instances of
 726 CIM_Accounts through instances of the CIM_AccountOnSystem associations from Host1.
- 727 2) The resource type is determined by the value of the ResourceType property of ACSSD1.
 728 Because the resource type is "File System", the administrator finds the instances of
 729 CIM_LogicalFiles representing the protected directories. The set of actions for FS1 can be
 730 determined by the instance of FSC1 as described in 9.3.
- 731 3) The administrator composes access control policies by assembling the determined information.

732 The format of the policy description is beyond the scope of this profile.

733 9.5 Determine policy transfer service

734 This subclause and the following subclauses show how the administrator distributes access policies
 735 edited as described in 9.4.



736

737

Figure 7 – Policy transfer service

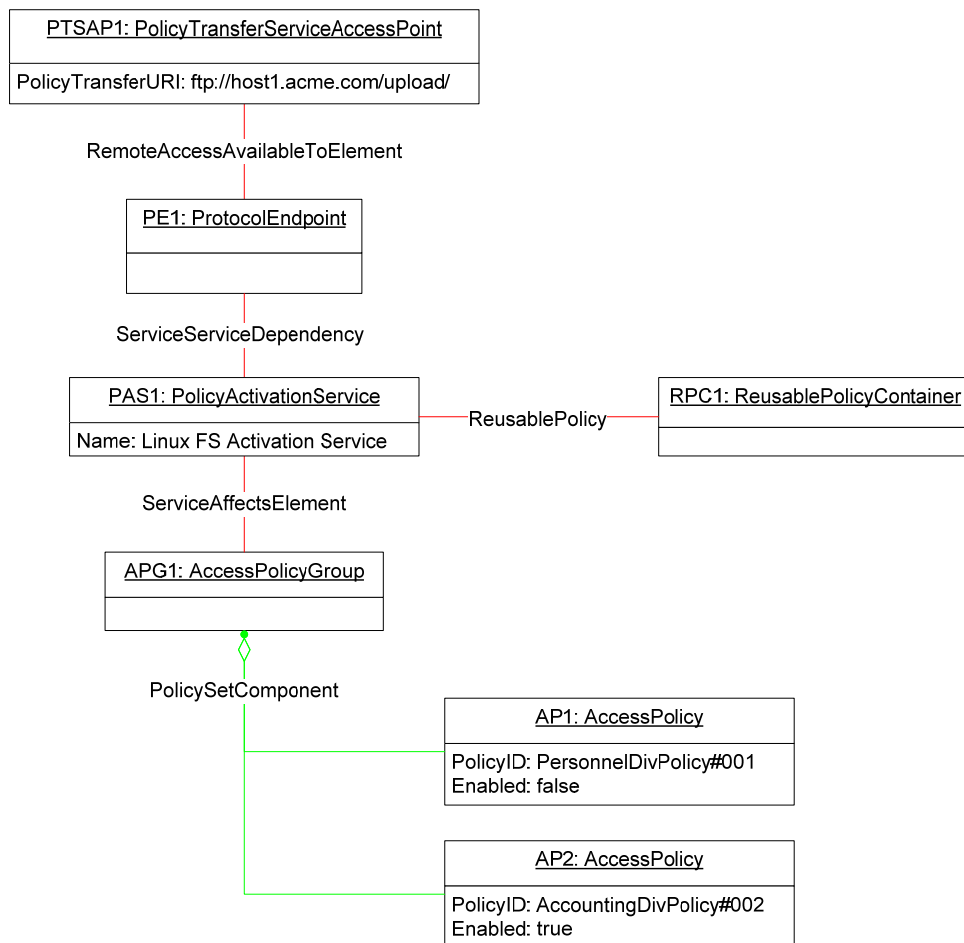
738 A policy transfer service is a network service used for distributing access control policies from a remote
 739 host; it can be represented by an instance of CIM_ProtocolEndpoint (PE1). The administrator on the
 740 remote host can identify address information (such as URI) of the service through an instance of

741 CIM_PolicyTransferServiceAccessPoint (PTSAP1) associated with PE1. In PTSAP1, the
 742 PolicyTransferURIs property provides a set of available URIs, such as an FTP service URI
 743 <ftp://host1.acme.com/upload/>.

744 The administrator on the remote host identifies address information as follows:

- 745 1) To find the CIM_PolicyTransferServiceAccessPoint instance, the administrator refers to the
 746 CIM_ComputerSystem instance (Host1), and then finds the CIM_AccessControlService
 747 instance (ACS1) through the CIM_HostedService association.
- 748 2) From ACS1, the administrator identifies the CIM_PolicyActivationService instance (PAS1)
 749 associated with the CIM_AssociatedPolicyActivationService association and then the
 750 CIM_ProtocolEndpoint instance (PE1) by traversing the CIM_ServiceServiceDependency
 751 association.
- 752 3) Through the CIM_RemoteAccessAvailableToElement association, the administrators can find
 753 the PolicyTransferServiceAccessPoint instance that collaborates with PAS1.

754 **9.6 Distribute access control policies**



755

756

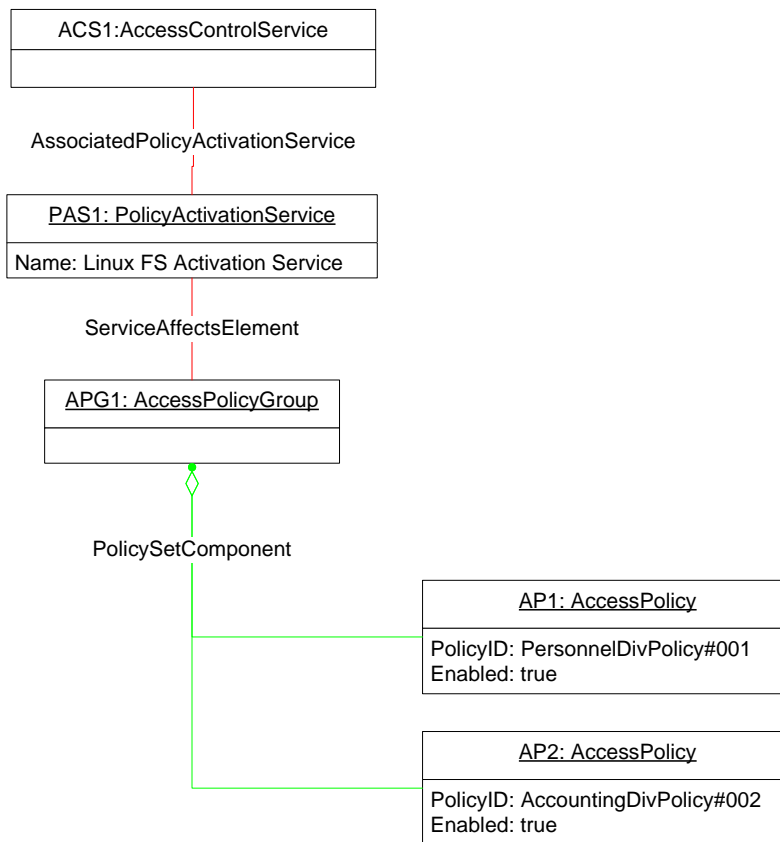
Figure 8 – Distributed access policies

757 The administrator transfers the access policies to the URL specified with the PolicyTransferURIs property
 758 of PTS1. In this case, because the URI is <ftp://host1.acme.com/upload>, the administrator transfers the
 759 policies with the FTP service.

760 When the administrator transfers two access policies, two CIM_AccessControlPolicy instances, AP1 and
 761 AP2, are instantiated. The PolicyID properties are initialized with unique identifiers; for example, the
 762 personnel division’s policy is “PersonnelDivPolicy#001” and the accounting division’s policy is
 763 “AccountingDivPolicy#002”. Also, the Enabled properties are set to *false*, because the policies are
 764 inactive until the consequent activation process.

765 The transferred policies are stored in a policy repository such as a specific directory or database, which is
 766 represented by an instance of CIM_ReusablePolicyContainer (RPC1). A group of the access control
 767 policies stored on the repository is represented by the CIM_AccessControlPolicyGroup instance (APG1).
 768 The RPC1 and APG1 instances are associated through the CIM_ReusablePolicy association. The
 769 transferred access policies AP1 and AP2 are aggregated to APG1 through the CIM_PolicySetComponent
 770 association.

771 **9.7 Activate access policies**



772

773

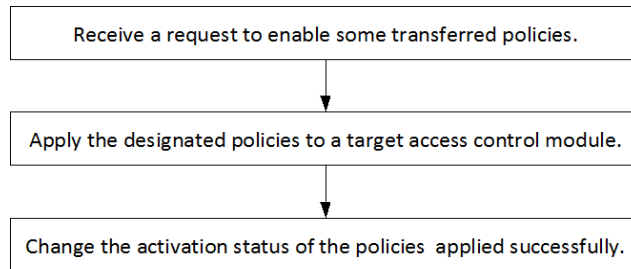
Figure 9 – Activated access policies

774 The administrator can activate the transferred policies AP1 and AP2 with the CIM_PolicyActivationService
 775 instance (PAS1) associated with the CIM_AccessControlService instance (ACS1) through the
 776 CIM_AssociatedPolicyActivationService association (see Figure 9).

777 In response to the activation request from the administrator, PAS1 performs the following steps:

- 778 1) It applies the policies to the associated access control services.
- 779 2) It changes to *true* the Enabled properties of both AP1 and AP2 (see Figure 10).

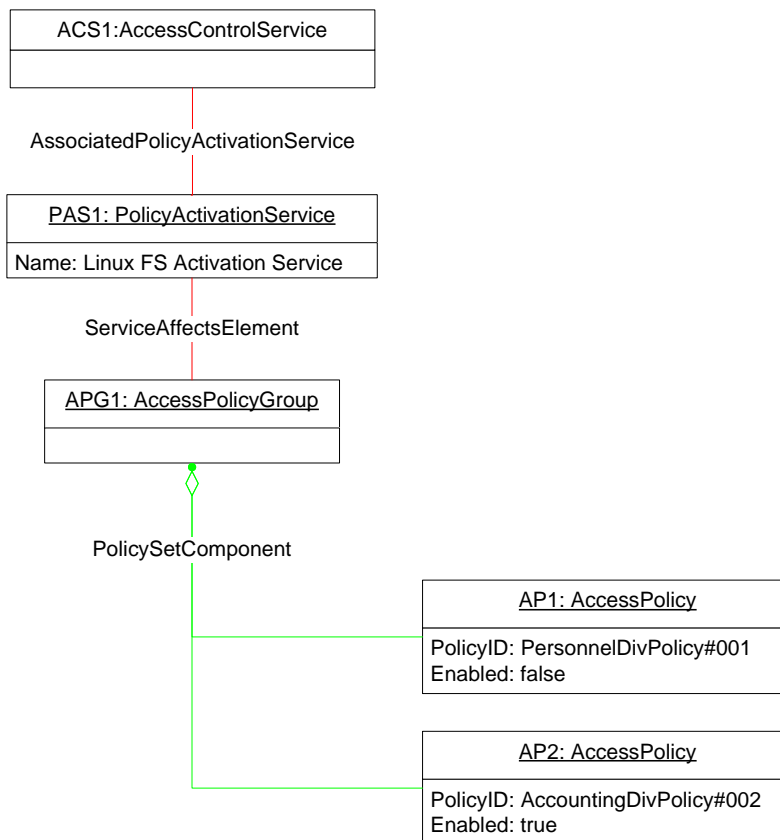
780 This activation operation is served as the ActivatePolicy() method.



781

782 **Figure 10 – Flow diagram of the policy activation process**

783 **9.8 Deactivate access policies**



784

785 **Figure 11 – Deactivated access policies**

786 Similarly, on a disabling request for AP1, PAS1 performs the following steps:
 787 1) It removes (or unloads) the policies from the target access control service.
 788 2) It changes the Enabled property of AP1 to false.
 789 This deactivation operation is served as the DeactivatePolicy() method of PAS1.
 790 On successful completion, the DeactivatePolicy() method may destroy the AP1 instance for some reason,
 791 for example, to reduce memory consumption.

792 **10 CIM Elements**

793 Table 16 lists CIM elements that are defined or specialized for this profile. Each CIM element shall be
 794 implemented as described in Table 16. The CIM Schema descriptions for any referenced element and its
 795 sub-elements apply.

796 Clauses 7 ("Implementation") and 8 ("Methods") may impose additional requirements on these elements.

797 **Table 16 – CIM Elements: Integrated Access Control Policy Management Profile**

Element Name	Requirement	Description
Classes		
CIM_HostedService	Mandatory	See 10.1.
CIM_AssociatedPolicyActivationService	Mandatory	See 10.2.
CIM_ElementSettingData	Mandatory	See 10.3.
CIM_ElementCapabilities	Mandatory	See 10.4.
CIM_PolicySetComponent	Mandatory	See 10.5.
CIM_ReusablePolicyContainer	Optional	See 10.6.
CIM_ReusablePolicy	Optional	See 10.7.
CIM_ServiceServiceDependency	Mandatory	See 10.8.
CIM_ServiceAffectsElement	Mandatory	See 10.9.
CIM_AccessControlService	Mandatory	See 10.10.
CIM_RegisteredProfile	Mandatory	See 10.11.
CIM_AccessControlServiceSettingData	Mandatory	See 10.12.
CIM_PolicyActivationService	Mandatory	See 10.13.
CIM_PolicyTransferServiceAccessPoint	Mandatory	See 10.14.
CIM_AccessControlPolicyGroup	Mandatory	See 10.15.
CIM_AccessControlPolicy	Mandatory	See 10.16.
CIM_FileSystemCapabilities	Conditional	See 10.17.
CIM_RelationalDatabaseCapabilities	Conditional	See 10.18.
CIM_DatabaseContainsTable	Conditional	See 10.19.
Indications		
None defined in this profile		

798 **10.1 CIM_HostedService**

799 CIM_HostedService is used to associate an instance of CIM_AccountManagementService,
 800 CIM_PolicyActivationService and CIM_AccessControlService with an instance of CIM_ComputerSystem
 801 that is the computer system hosting the service. Table 17 contains the requirements for elements of this
 802 class.

803 **Table 17 – Class: CIM_HostedService**

Elements	Requirement	Description
Antecedent	Mandatory	Key: This property shall reference the instance of CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	Key: This property shall reference the instance of CIM_AccountManagementService, CIM_PolicyActivationService, or CIM_AccessControlService. Cardinality 1..*

804 **10.2 CIM_AssociatedPolicyActivationService**

805 CIM_AssociatedPolicyActivationService is used to associate an instance of CIM_PolicyActivationService
 806 with an instance of CIM_AccessControlService that represents the access control service controlled
 807 through the activation service. Table 18 contains the requirements for elements of this class.

808 **Table 18 – Class: CIM_AssociatedPolicyActivationService**

Elements	Requirement	Description
ActivationService	Mandatory	Key: This property shall reference the instance of CIM_PolicyActivationService. Cardinality *
AccessControlService	Mandatory	Key: This property shall reference the instance of CIM_AccessControlService. Cardinality 1..*

809 **10.3 CIM_ElementSettingData**

810 CIM_ElementSettingData is used to associate an instance of CIM_AccessControlService with an instance
 811 of CIM_AccessControlServiceSettingData that represents the configurations of the access control service.
 812 Table 19 contains the requirements for elements of this class.

813 **Table 19 – Class: CIM_ElementSettingData**

Elements	Requirement	Description
ManagedElement	Mandatory	Key: This property shall reference the instance of CIM_AccessControlService. Cardinality 1
SettingData	Mandatory	Key: This property shall reference the instance of CIM_AccessControlServiceSettingData. Cardinality 1..*

814 **10.4 CIM_ElementCapabilities**

815 CIM_ElementCapabilities is used to associate an instance of CIM_ManagedElement with an instance of
 816 CIM_Capabilities that represents the action types corresponding to the resource type. Table 20 contains
 817 the requirements for elements of this class.

818 **Table 20 – Class: CIM_ElementCapabilities**

Elements	Requirement	Description
ManagedElement	Mandatory	Key: This property shall reference the instance of CIM_ManagedElement. Cardinality 1..*
Capabilities	Mandatory	Key: This property shall reference the instance of CIM_Capabilities. Cardinality 1..*

819 **10.5 CIM_PolicySetComponent**

820 CIM_PolicySetComponent is used to aggregate the instances of CIM_AccessControlPolicy into an
 821 instance of CIM_AccessControlPolicyGroup, which represents a set of access control policies distributed
 822 through policy transfer services. Table 21 contains the requirements for elements of this class.

823 **Table 21 – Class: CIM_PolicySetComponent**

Elements	Requirement	Description
GroupComponent	Mandatory	Key: This property shall reference the instance of CIM_AccessControlPolicyGroup. Cardinality 1
PartComponent	Mandatory	Key: This property shall reference the instance of CIM_AccessControlPolicy. Cardinality 1..*

824 **10.6 CIM_ReusablePolicyContainer**

825 CIM_ReusablePolicyContainer is an optional class that represents a policy repository containing the
 826 access control policies. Table 22 contains the requirements of elements of this class.

827 **Table 22 – Class: CIM_ReusablePolicyContainer**

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

828 **10.7 CIM_ReusablePolicy**

829 CIM_ReusablePolicy is an optional association that is used to associate an instance of
 830 CIM_AccessControlPolicyGroup with an instance of CIM_ReusablePolicyContainer. Table 23 contains
 831 the requirements for elements of this class.

832

Table 23 – Class: CIM_ReusablePolicy

Elements	Requirement	Description
Antecedent	Mandatory	Key: This property shall reference the instance of CIM_ReusablePolicyContainer. Cardinality 1..*
Dependent	Mandatory	Key: This property shall reference the instance of CIM_AccessControlPolicyGroup. Cardinality 1

833 **10.8 CIM_ServiceServiceDependency**

834 CIM_ServiceServiceDependency is used to associate an instance of CIM_PolicyActivationService with an
 835 instance of CIM_ProtocolEndpoint that represents an endpoint of a policy transfer service used for access
 836 policy distribution. Table 24 contains the requirements for elements of this class.

837

Table 24 – Class: CIM_ServiceServiceDependency

Elements	Requirement	Description
Antecedent	Mandatory	Key: This property shall reference the instance of CIM_PolicyActivationService. Cardinality 1..*
Dependent	Mandatory	Key: This property shall reference the instance of CIM_ProtocolEndpoint. Cardinality 1..*
TypeOfDependency	Mandatory	Matches 5 (Cooperate)

838 **10.9 CIM_ServiceAffectsElement**

839 CIM_ServiceAffectsElement is an optional association class that is used to associate an instance of
 840 CIM_ComputerSystem with an instance of CIM_ManagedElement that represents the resource to protect
 841 by the access control service. Table 25 contains the requirements for elements of this class.

842

Table 25 – Class: CIM_ServiceAffectsElement

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

843 **10.10 CIM_AccessControlService**

844 CIM_AccessControlService is used to represent the access control service installed on a certain
 845 computer system. Table 26 contains the requirements for elements of this class.

846

Table 26 – Class: CIM_AccessControlService

Elements	Requirement	Description
CreationClassName	Mandatory	Key
Name	Mandatory	Key
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
ImplementationType	Optional	See 7.1.3.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “. *”).
Version	Mandatory	Key

847 **10.11 CIM_RegisteredProfile**

848 The CIM_RegisteredProfile class is defined by the [Profile Registration Profile](#). The requirements denoted
849 in Table 27 are in addition to those mandated by the [Profile Registration Profile](#).

850

Table 27 – Class: CIM_RegisteredProfile

Elements	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Integrated Access Control Policy Management”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

851 **10.12 CIM_AccessControlServiceSettingData**

852 CIM_AccessControlServiceSettingData is used to represent the type of access control supported by the
853 access control service associated through an instance of CIM_ElementSettingData. Table 28 contains the
854 requirements for elements of this class.

855

Table 28 – Class: CIM_AccessControlServiceSettingData

Elements	Requirement	Description
InstanceID	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “. *”).
PrincipalType	Mandatory	See 7.2.
ResourceType	Mandatory	See 7.3.

856 **10.13 CIM_PolicyActivationService**

857 CIM_PolicyActivationService is used to represent a service to enable and disable the distributed policies
858 on a target access control service. Table 29 contains the requirements for elements of this class.

859

Table 29 – Class: CIM_PolicyActivationService

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").
ActivatePolicy()	Mandatory	See 8.1.1.
DeactivatePolicy()	Mandatory	See 8.1.2.

860 **10.14 CIM_PolicyTransferServiceAccessPoint**

861 CIM_PolicyTransferServiceAccessPoint is used to represent a service to transfer access control policies
 862 to the target access control service. Table 30 contains the requirements for elements of this class.

863

Table 30 – Class: CIM_PolicyTransferServiceAccessPoint

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
PolicyTransferURIs	Mandatory	See 7.4.2.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

864 **10.15 CIM_AccessControlPolicyGroup**

865 CIM_AccessControlPolicyGroup is used to represent a set of access control policies for a certain access
 866 control service on the target computer system. Table 31 contains the requirements for elements of this
 867 class.

868

Table 31 – Class: CIM_AccessControlPolicyGroup

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyGroupName	Mandatory	Key
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern ".*").

869 **10.16 CIM_AccessControlPolicy**

870 CIM_AccessControlPolicy is used to represent an access policy distributed to the target computer system.
 871 Table 32 contains the requirements for elements of this class.

872

Table 32 – Class: CIM_AccessControlPolicy

Elements	Requirement	Description
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
PolicyID	Mandatory	Key (see 7.5.2)
Enabled	Mandatory	See 7.5.3.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “..*”).

873 **10.17 CIM_FileSystemCapabilities**

874 CIM_FileSystemCapabilities is specialized to represent supported operations to associated file systems.
875 Table 33 contains the requirements for elements of this class.

876

Table 33 – Class: CIM_FileSystemCapabilities

Elements	Requirement	Description
InstanceID	Mandatory	Key
SupportedOperations	Conditional	See 7.3.2.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “..*”).

877 **10.18 CIM_RelationalDatabaseCapabilities**

878 CIM_RelationalDatabaseCapabilities is used to represent supported operations to associated databases.
879 Table 34 contains the requirements for elements of this class.

880

Table 34 – Class: CIM_RelationalDatabaseCapabilities

Elements	Requirement	Description
InstanceID	Mandatory	Key
SupportedDBOperations	Conditional	See 7.3.4.
SupportedTableOperations	Conditional	See 7.3.4.
SupportedColumnOperations	Conditional	See 7.3.4.
ElementName	Mandatory	This property shall be formatted as a free-form string of variable length (pattern “..*”).

881 **10.19 CIM_DatabaseContainsTable**

882 CIM_DatabaseContainsTable is an association class to be used for associating CIM_CommonDatabase
883 instances with CIM_SqlTable ones. Table 35 contains the requirements for elements of this class.

884

Table 35 – Class: CIM_DatabaseContainsTable

Elements	Requirement	Description
Antecedent	Mandatory	Key: This property shall reference the instance of CIM_CommonDatabase. Cardinality 1
Dependent	Mandatory	Key: This property shall reference the instance of CIM_SqlTable. Cardinality *

885

886
887
888
889
890

**ANNEX A
(informative)**

Change Log

Version	Date	Description
1.0.0a	2011-09-16	DMTF Standard

891