Document Number: DSP1106

Date: 2011-09-16

Version: 1.0.0

Integrated Access Control Policy Management Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

Contents

Figures

Tables

134

Foreword

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

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

Acknowledgments

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

- Mark Carlson – Sun
- Jorge Lobo – IBM
- Fumio Machida – NEC
- Masayuki Nakae – NEC
- Ryuichi Ogawa – NEC

Introduction

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

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

Document conventions

Typographical conventions

The following typographical conventions are used in this document:

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

ABNF usage conventions

Format definitions in this document are specified using ABNF (see RFC5234), with the following deviations:

- Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the definition in RFC5234 that interprets literal strings as case-insensitive US-ASCII characters.

Integrated Access Control Policy Management Profile

1 Scope

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

2 Normative references

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

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

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

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

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

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

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

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

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

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

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

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

3 Terms and definitions

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

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

The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

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

The terms defined in DSP0004, DSP0200, and DSP1001 apply to this document. The following additional terms are used in this document.

3.1

access control policy activation service

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

3.2

access control policy composition

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

3.3

access control policy distribution

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

3.4

access control policy group

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

3.5

access control policy transfer service access point

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

3.6

access control service

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

3.7

action type

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

3.8
activation status
a status of access policy activation process

3.9
principal type
a class of principals supported by a certain access control service (for example, users, accounts, groups)

3.10
resource type
a class of target objects supported by a certain access control service (for example, files, virtual machines, data bases)

4 Symbols and abbreviated terms

The following abbreviations are used in this document.

4.1
IAM
Integrated Access Control Manager

4.2
URI
universal resource identifier

4.3
XACML
Extensible Access Control Markup Language

5 Synopsis

Profile Name: Integrated Access Control Policy Management

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.19

Central Class: CIM_AccessControlService

Scoping Class: CIM_ComputerSystem

The *Integrated Access Control Policy Management Profile* provides the ability to compose and distribute common access control policies for different access control components in managed systems.

The Central Class of the *Integrated Access Control Policy Management Profile* shall be CIM_AccessControlService. The Central Instance shall be an instance of CIM_AccessControlService. The Scoping Class shall be CIM_ComputerSystem. The Scoping Instance shall be the instance of CIM_ComputerSystem associated to the Central Instance through the CIM_HostedService association.

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.

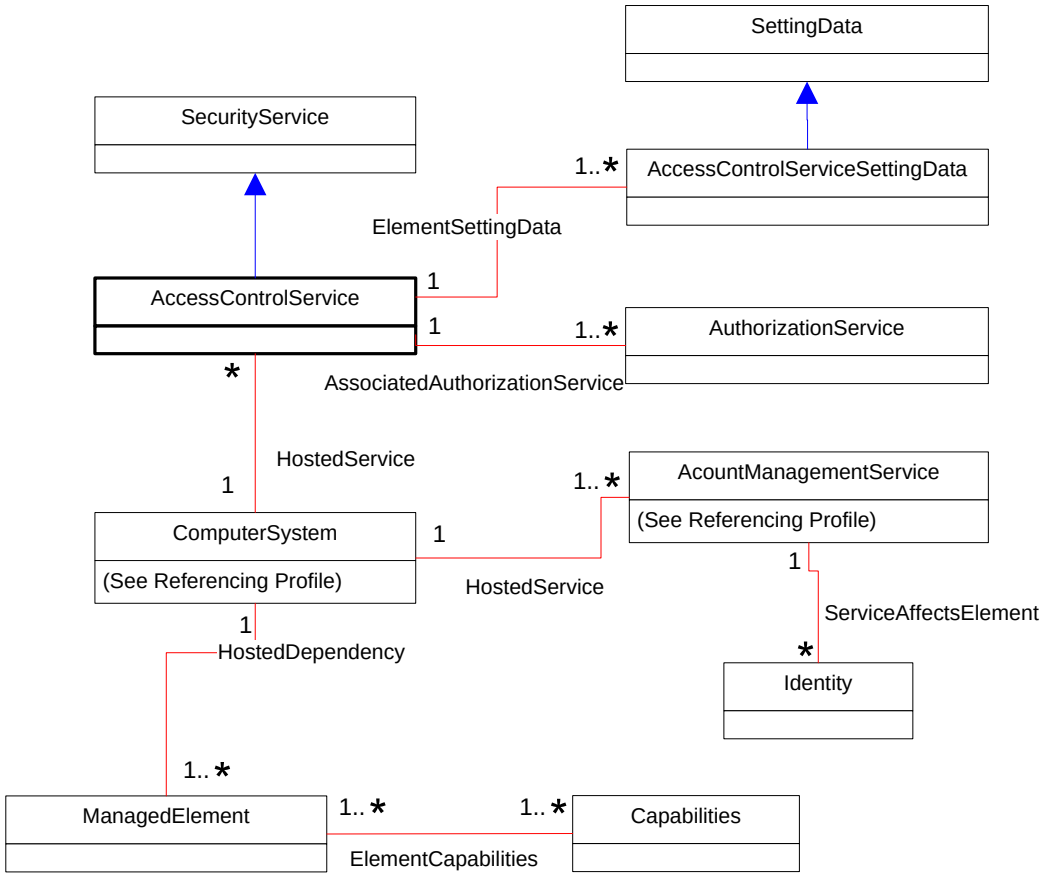


Figure 1 – Access control policy composition: Class diagram

6.1.2 Access control service

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

6.1.3 Principal type and resource type

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

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

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

6.1.4 Resource-related capabilities

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

6.2 Access control policy distribution

6.2.1 General

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

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

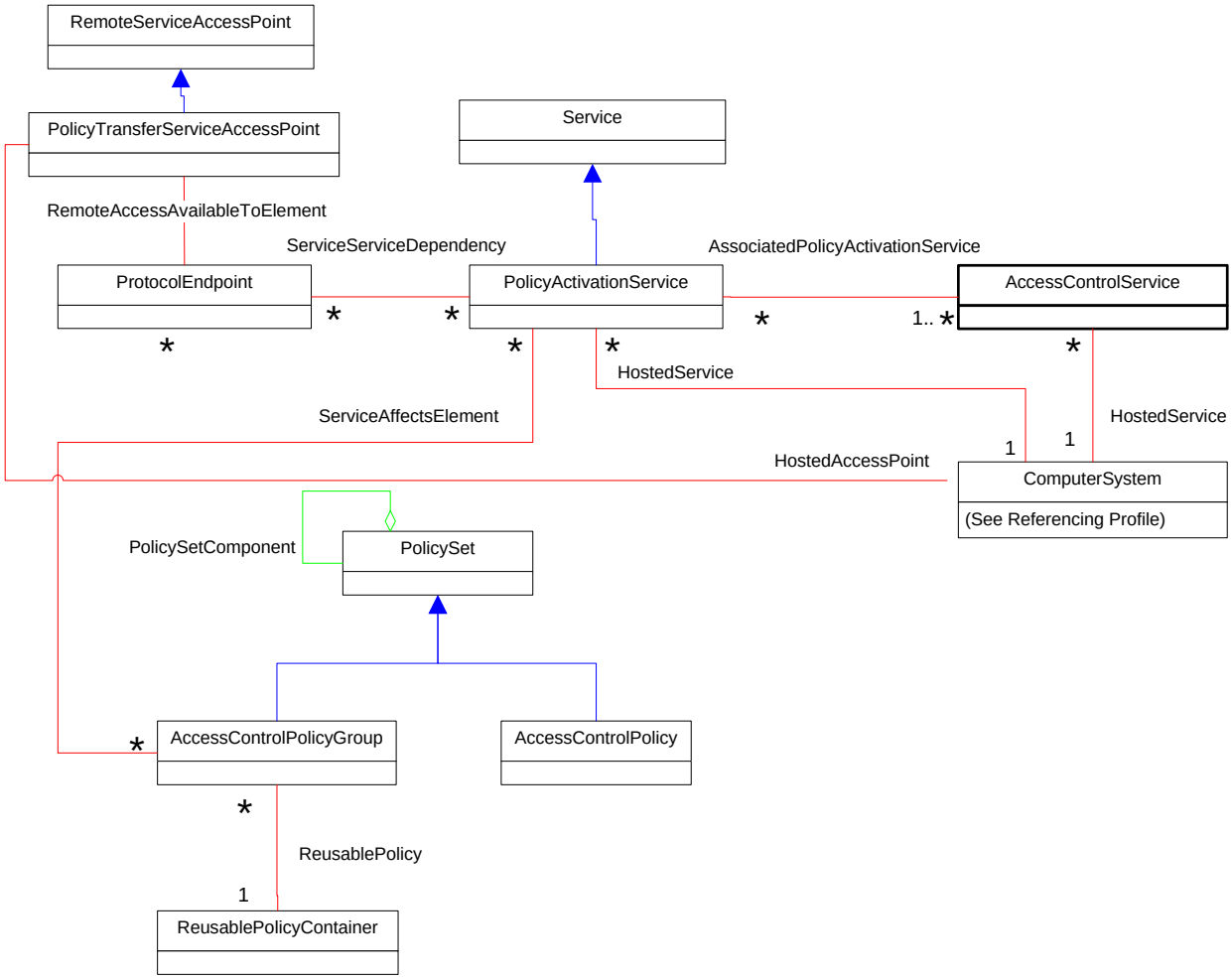


Figure 2 – Access control policy distribution: Class diagram

6.2.2 Policy transfer service access point

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

6.2.3 Access control policy

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

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

6.2.4 Policy activation service

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

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

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

7 Implementation

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

7.1 Access control service

7.1.1 General

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

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

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

7.1.2 Version

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

7.1.3 Implementation type (optional)

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

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

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

7.2 Principals

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

7.3 Resources and resource-related capabilities

7.3.1 General

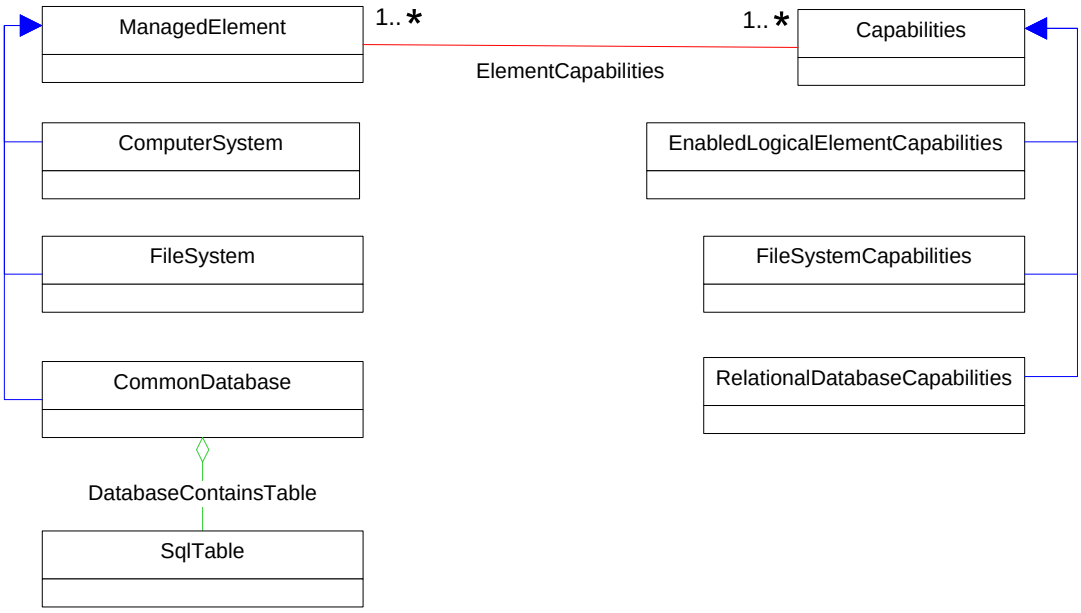
All resources that are managed by the specific access control service shall be represented by instances of CIM_ManagedElement subclasses. The resource type is specified in the ResourceType property of the CIM_AccessControlServiceSettingData instance. The system administrator finds the instances of CIM_ManagedElement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

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

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

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

- Virtual machine: CIM_ComputerSystem and CIM_EnabledLogicalElementCapabilities
- File system: CIM_FileSystem and CIM_FileSystemCapabilities
- 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

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

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

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

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

7.4 Policy transfer service access point

7.4.1 General

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

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

7.4.2 Policy transfer URI

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

7.5 Access control policy

7.5.1 General

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

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

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

7.5.2 Policy identifier

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

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

7.5.3 Activation status

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

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

7.6 Policy activation service

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

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

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

8 Methods

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

8.1 Extrinsic methods

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

8.1.1 CIM_PolicyActivationService.ActivatePolicy( )

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

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

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

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

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

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

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

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

Value	Description
0	Activation completed successfully.
1	Failed

8.1.2 CIM_PolicyActivationService.DeactivatePolicy( )

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

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

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

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

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

The DeactivatePolicy( ) method's parameters are specified in Table 4, and its return codes are specified in Table 5.

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

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

Value	Description
0	Deactivation completed successfully.
1	Failed

8.2 Profile conventions for operations

8.2.1 General

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

The default list of operations is as follows:

- GetInstance
- Associators
- AssociatorNames
- References
- ReferenceNames
- EnumerateInstances
- EnumerateInstanceNames

8.2.2 CIM_HostedService

Table 6 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 6, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 6 – Operations: CIM_HostedService

Operation	Requirement	Messages
Associators	Unspecified	None

AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.2.3 CIM_AssociatedPolicyActivationService

Table 7 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 7, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 7 – Operations: CIM_AssociatedPolicyActivationService

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

8.2.4 CIM_ElementSettingData

Table 8 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 8, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 8 – Operations: CIM_ElementSettingData

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

8.2.5 CIM_ElementCapabilities

Table 9 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 9, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 9 – Operations: CIM_ElementCapabilities

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None

References	Unspecified	None
ReferenceNames	Unspecified	None

8.2.6 CIM_PolicySetComponent

Table 10 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 10, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 10 – Operations: CIM_PolicySetComponent

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

8.2.7 CIM_ReusablePolicy

Table 11 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 11, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 11 – Operations: CIM_ReusablePolicy

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

8.2.8 CIM_ServiceServiceDependency

Table 12 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 12, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 12 – Operations: CIM_ServiceServiceDependency

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

ReferenceNames	Unspecified	None

8.2.9 CIM_HostedAccessPoint

Table 13 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 13, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 13 – Operations: CIM_HostedAccessPoint

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

8.2.10 CIM_ServiceAffectsElement

Table 14 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 14, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 14 – Operations: CIM_ServiceAffectsElement

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

8.2.11 CIM_AccessControlService

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.12 CIM_AccessControlServiceSettingData

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.13 CIM_PolicyActivationService

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.14 CIM_PolicyTransferServiceAccessPoint

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.15 CIM_AccessControlPolicyGroup

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.16 CIM_AccessControlPolicy

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.17 CIM_ProtocolEndpoint

All operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

8.2.18 CIM_RemoteAccessAvailableToElement

Table 15 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 15, all operations in the default list in 8.2.1 shall be implemented as defined in DSP0200.

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

Table 15 – Operations: CIM_ServiceAffectsElement

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

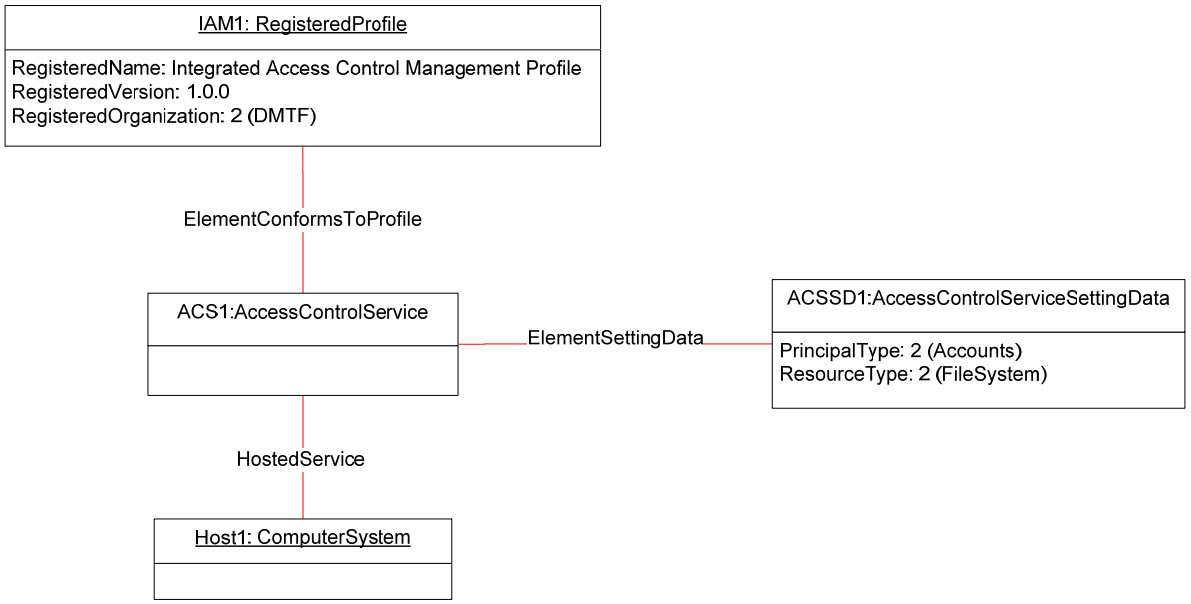
9 Use cases

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

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

9.1 Discover conformant access control service

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

9.3 Determine the resource related capabilities

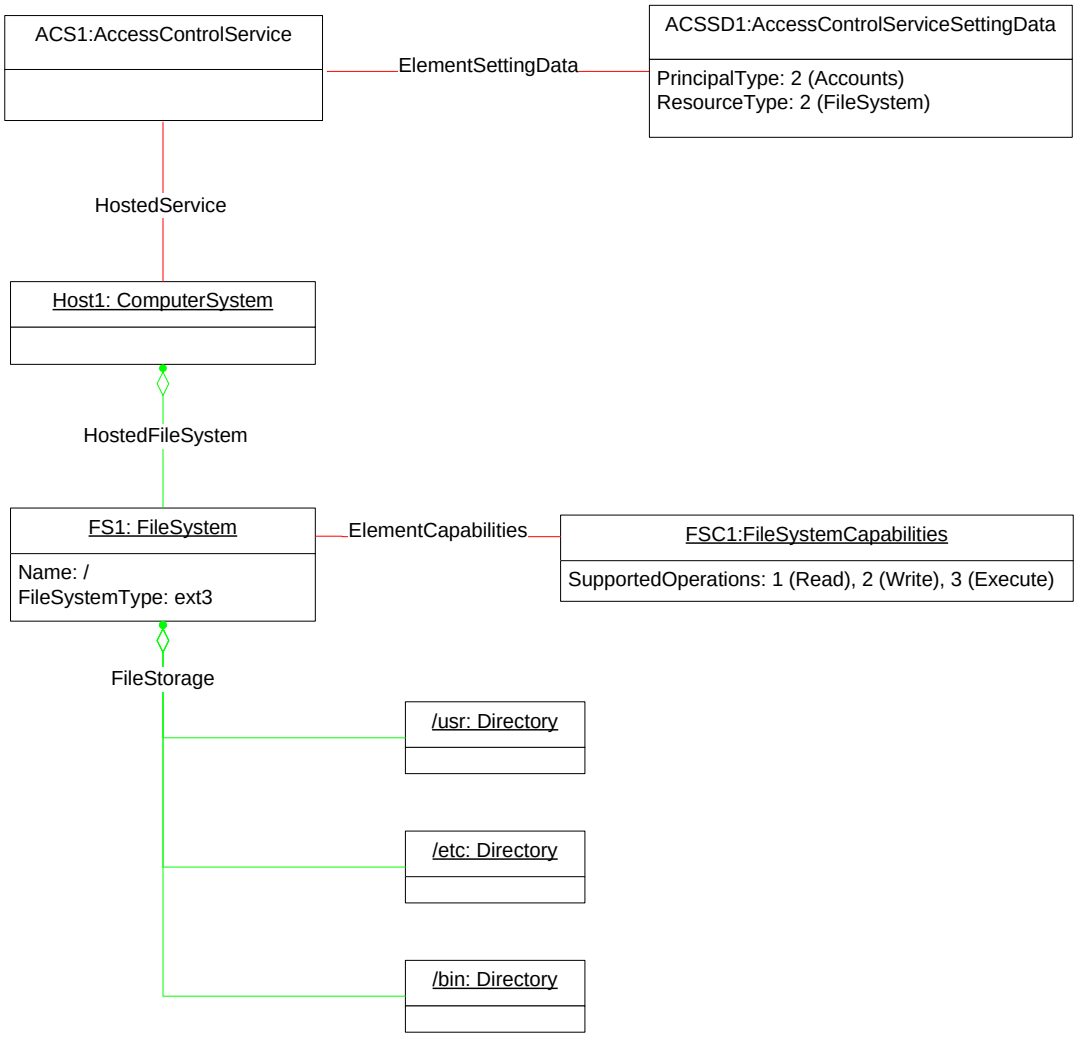


Figure 5 – Hosted file system and related capabilities

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

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

9.3.1 Other resource types

9.3.1.1 Virtual machines

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

9.3.1.2 Relational databases

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

9.4 Compose access control policies

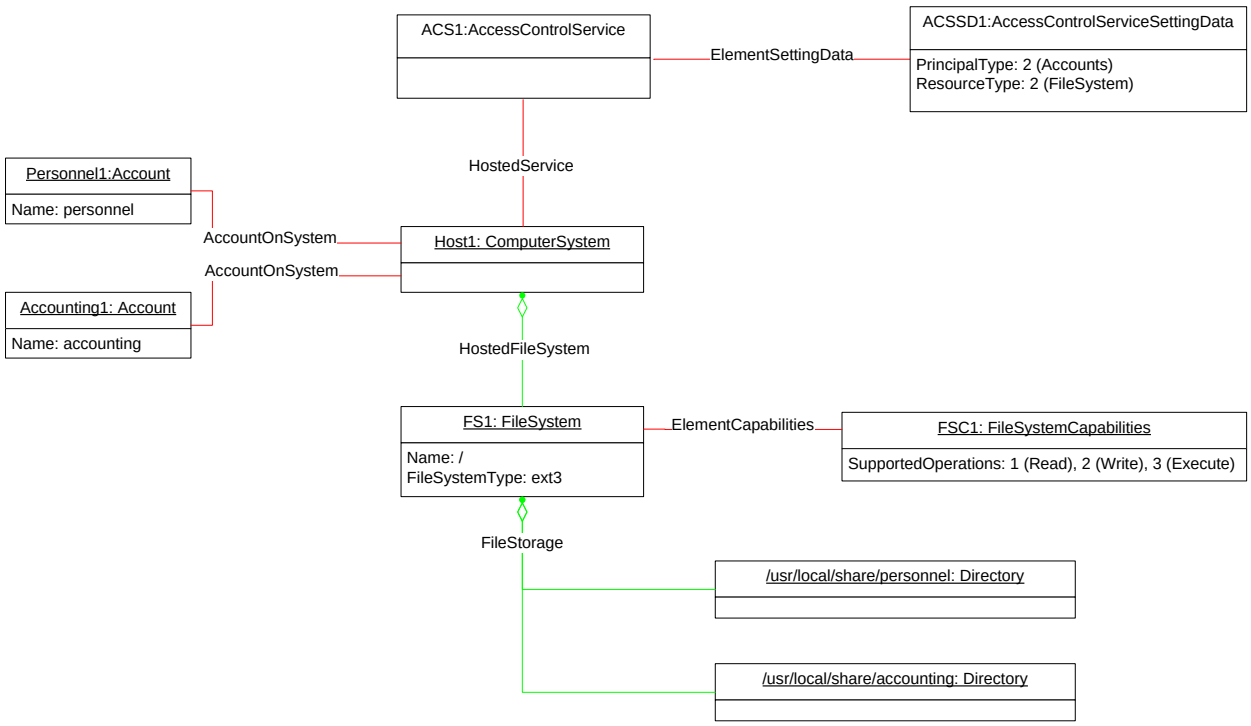


Figure 6 – Accounts and resources for policy composition

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

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

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

1) The type of principal supported by RM1 is determined by ACSSD1. Because the value of the PrincipalType property is "Accounts", the system administrator finds the instances of CIM_Accounts through instances of the CIM_AccountOnSystem associations from Host1.

2) The resource type is determined by the value of the ResourceType property of ACSSD1. Because the resource type is "File System", the administrator finds the instances of CIM_LogicalFiles representing the protected directories. The set of actions for FS1 can be determined by the instance of FSC1 as described in 9.3.

3) The administrator composes access control policies by assembling the determined information.

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

9.5 Determine policy transfer service

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

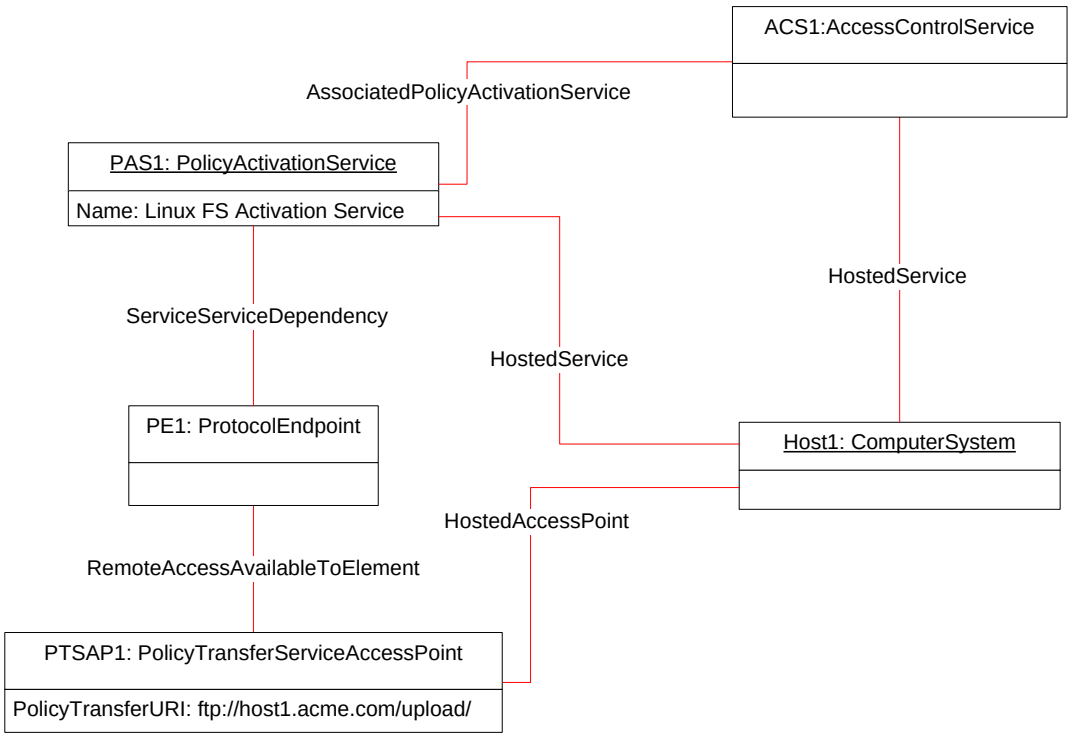


Figure 7 – Policy transfer service

A policy transfer service is a network service used for distributing access control policies from a remote host; it can be represented by an instance of CIM_ProtocolEndpoint (PE1). The administrator on the 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/>.

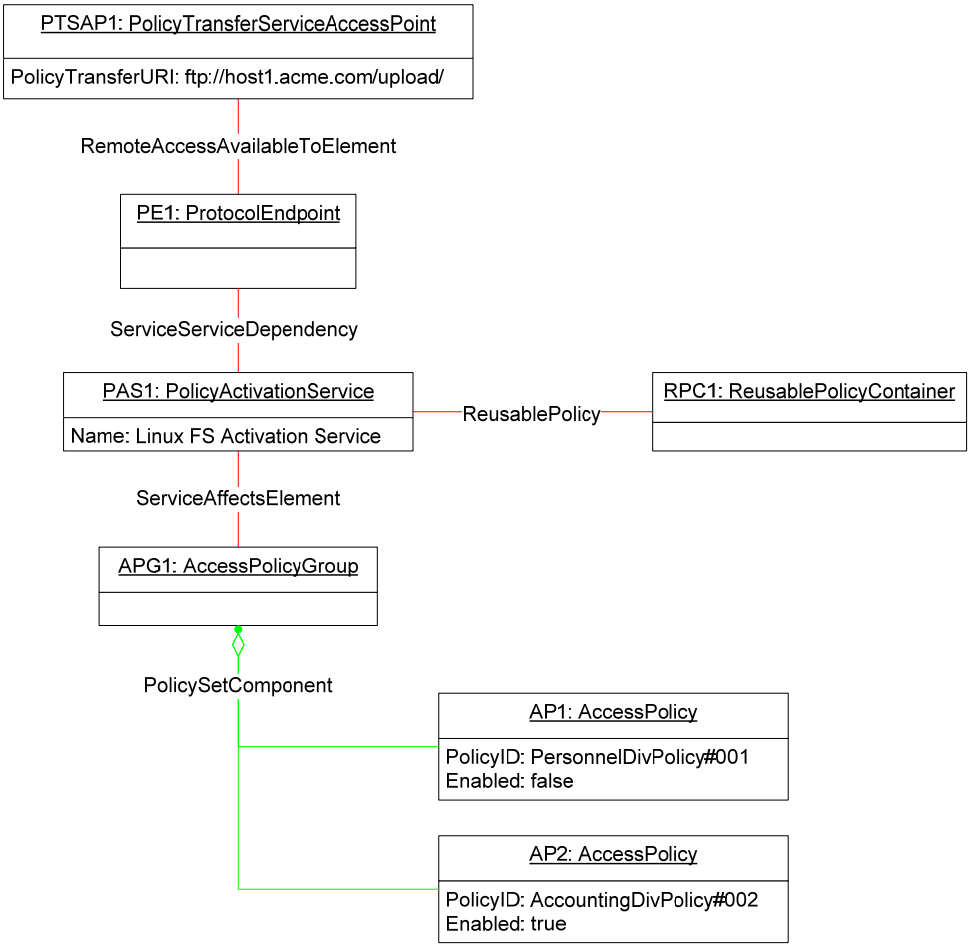
744 The administrator on the remote host identifies address information as follows:

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**

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

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

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

9.7 Activate access policies

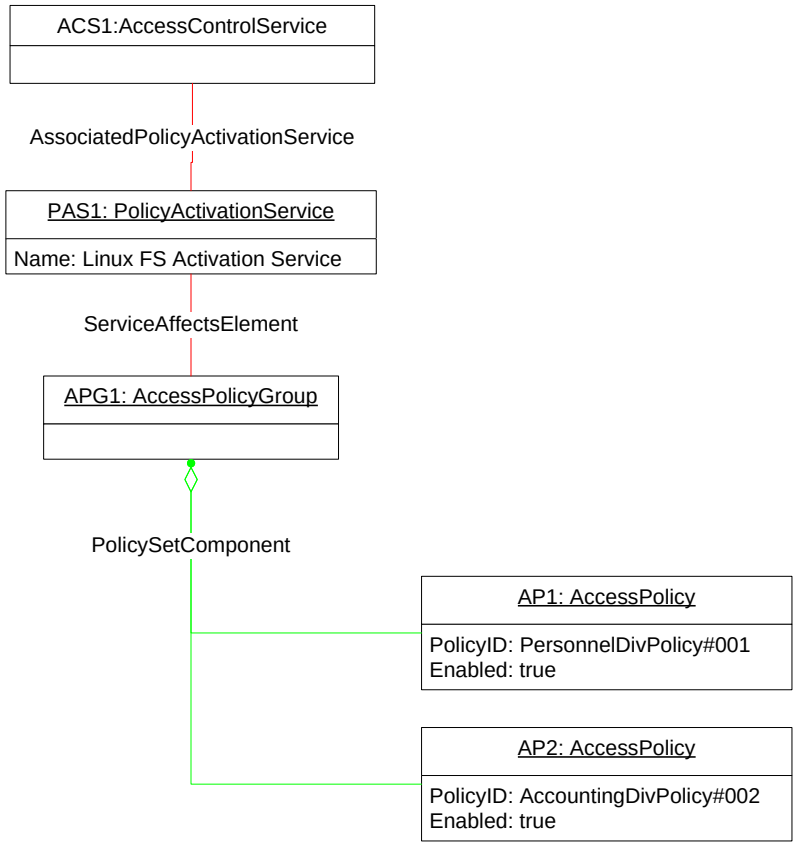


Figure 9 – Activated access policies

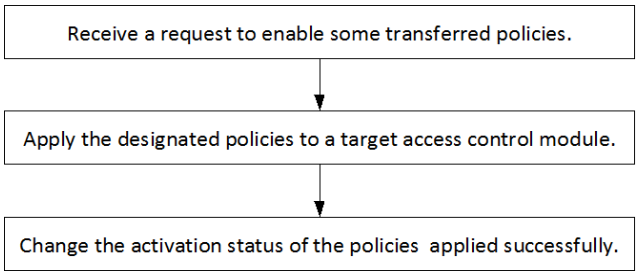
The administrator can activate the transferred policies AP1 and AP2 with the CIM_PolicyActivationService instance (PAS1) associated with the CIM_AccessControlService instance (ACS1) through the CIM_AssociatedPolicyActivationService association (see Figure 9).

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).

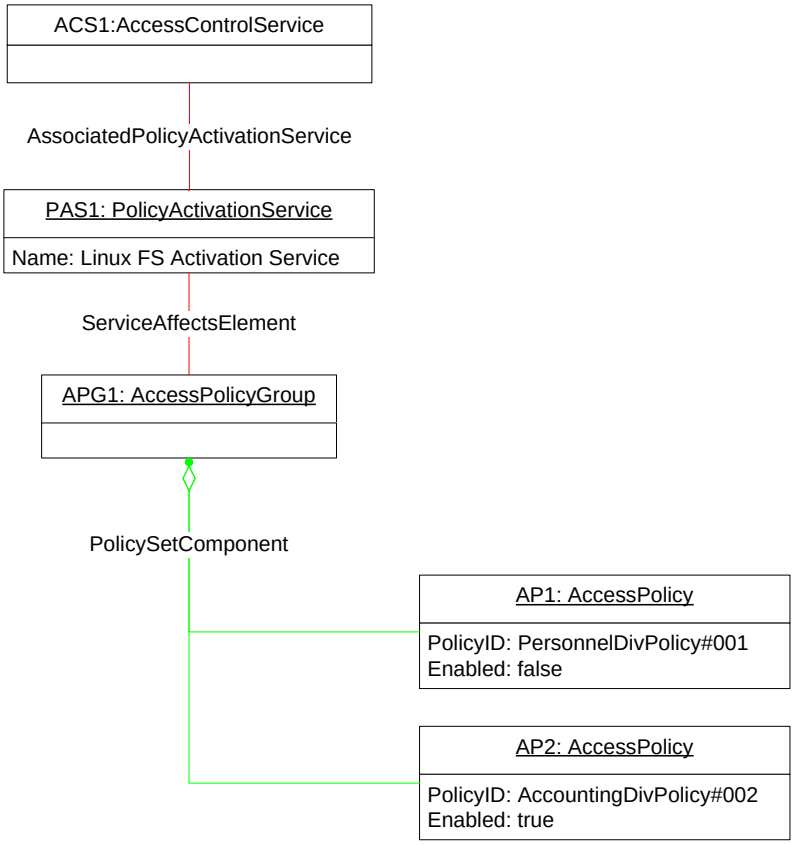
780 This activation operation is served as the ActivatePolicy( ) method.



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		

10.1 CIM_HostedService

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

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..*

10.2 CIM_AssociatedPolicyActivationService

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

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..*

10.3 CIM_ElementSettingData

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

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..*

10.4 CIM_ElementCapabilities

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

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..*

10.5 CIM_PolicySetComponent

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

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..*

10.6 CIM_ReusablePolicyContainer

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

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 ".*").

10.7 CIM_ReusablePolicy

CIM_ReusablePolicy is an optional association that is used to associate an instance of CIM_AccessControlPolicyGroup with an instance of CIM_ReusablePolicyContainer. Table 23 contains 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

ANNEX A
(informative)

Change Log

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