Document Number: DSP1039

Date: 2008-10-31

Version: 1.0.0

Role Based Authorization Profile

Document Type: Specification

Document Status: Final

Document Language: E

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

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

Contents

Figures

Tables

Foreword

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

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

Introduction

This document defines the classes used to describe role-based authorization in a managed system. Also included are descriptions of the relationship between the authorization and authentication for a managed system, and the DMTF profile version information. 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 that represent the component described in this document.

Role Based Authorization Profile

1 Scope

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

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.

2.1 Approved References

DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

DMTF DSP1034, *Simple Identity Management Profile 1.0*

DMTF DSP1033, *Profile Registration Profile 1.0*

2.2 References under Development

DMTF DSP0215, *Server Management Managed Element Addressing Specification, 1.0.0*

2.3 Other References

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

Unified Modeling Language (UML) from the Open Management Group (OMG), http://www.uml.org

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. For the purposes of this document, the terms and definitions given in DSP1033 and DSP1001 also apply.

3.1

can

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

3.2

cannot

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

3.3

conditional

indicates requirements to be followed strictly to conform to the document when the specified conditions are met

3.4

mandatory

indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.5

may

indicates a course of action permissible within the limits of the document

3.6

need not

indicates a course of action permissible within the limits of the document

3.7

optional

indicates a course of action permissible within the limits of the document

3.8

referencing profile

indicates a profile that owns the definition of this class and can include a reference to this profile in its “Referenced Profiles” table

3.9

shall

indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.10

shall not

indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.11

should

indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12

should not

indicates that a certain possibility or course of action is deprecated but not prohibited

3.13

unspecified

indicates that this profile does not define any constraints for the referenced CIM element or operation

3.14

Associated Privilege Management Capability

an instance of CIM_RoleBasedManagementCapabilities describing the capabilities of the mentioned instance of CIM_Privilege as described in section 7.4

3.15

Associated Role Management Capability

an instance of CIM_RoleBasedManagementCapabilities, which is associated with the instance of CIM_RoleBasedAuthorizationService through the CIM_ElementCapabilities association, which in turn is associated with the mentioned instance of CIM_Role through the CIM_ServiceAffectsElement association

3.16

Cumulative Privilege

a conceptual instance of CIM_Privilege that represents rights granted

3.17

Cumulative Role Privilege

an instance of CIM_Privilege that is the conceptual representation of all the Granted Privileges and Denied Privileges that are associated with a particular instance of CIM_Role

3.18

Denied Privilege

an instance of CIM_Privilege with the PrivilegeGranted property set to FALSE that represents the denied privilege of associated roles

3.19

Granted Privilege

an instance of CIM_Privilege with the PrivilegeGranted property set to TRUE that represents the granted privilege of associated roles

3.20

Modified Role

an instance of CIM_Role that is referenced by the Role parameter of the ModifyRole( ) method

3.21

Root Instance

an instance of CIM_ManagedElement that is associated with the instance of CIM_Role through the CIM_RoleLimitedToTarget association and conceptually symbolizes the root of the scope hierarchy for the CIM_Role instance

3.22

Template Privilege

an instance of CIM_Privilege only to be used by a client as a template for creating new authorized roles or modifying the existing roles

4 Symbols and Abbreviated Terms

Experimental Maturity Level

Some of the content considered for inclusion in the *Role Based Authorization Profile* has yet to receive sufficient review to satisfy the adoption requirements set forth by the Technical Committee within the DMTF. This content is presented here as an aid to implementers who are interested in likely future developments within this specification. The content marked experimental may change as implementation experience is gained. There is a high likelihood that it will be included in an upcoming revision of the specification. Until that time, it is purely informational, and is clearly marked within the text.

A sample of the typographical convention for experimental content is included here:

EXPERIMENTAL

Experimental content appears here.

EXPERIMENTAL

5 Synopsis

Profile Name: *Role Based Authorization*

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.20

Central Class: CIM_RoleBasedAuthorizationService

Scoping Class: CIM_ComputerSystem

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

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

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

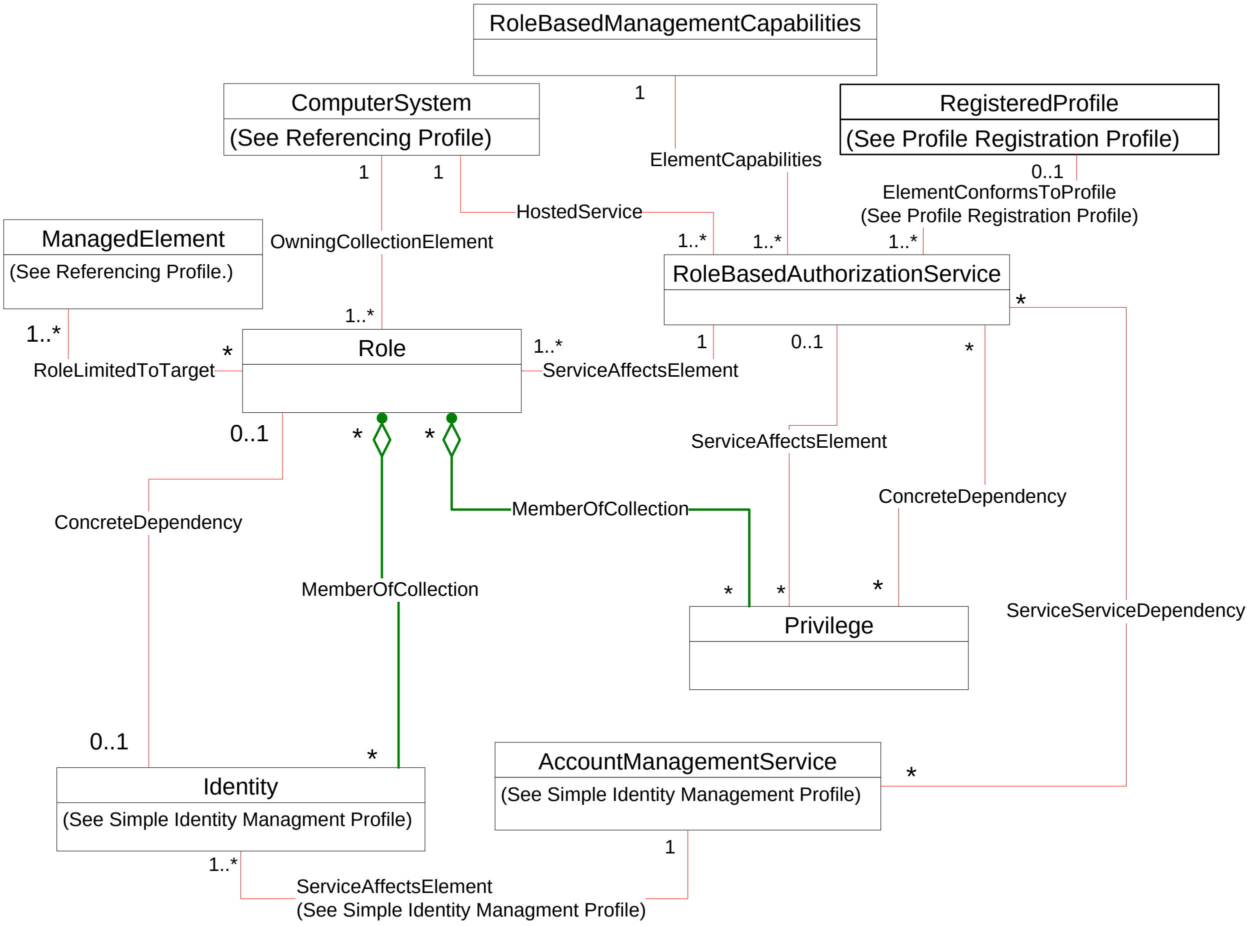
Table 1 – Referenced Profiles

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

6 Description

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

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



321

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

323 6.1 Role Authorization Service: CIM_RoleBasedAuthorizationService

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

329 6.2 Authorized Roles and Privileges: CIM_Role and CIM_Privilege

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

6.2.1 Role Privileges

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

6.2.2 Role Scope

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

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

6.2.3 Cumulative Privileges

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

6.3 Security Principal: CIM_Identity

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

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

6.4 Privilege Management

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

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

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

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

7 Implementation

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

7.1 Modeling the Authorized Role

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

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

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

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

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

7.1.1 Scope of the Authorized Role

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

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

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

Table 2 – Containment Relationships

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

7.1.1.1 Managed Element within Role's Scope

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

An instance of CIM_ManagedElement shall be in the scope of an instance of CIM_Role if

1) The instance of CIM_ManagedElement is associated with the instance of CIM_Role through the CIM_RoleLimitedToTarget association.

2) The instance of CIM_ManagedElement is referenced by an instance of an association class specified in the "Association Class" column of Table 2 where a reference to the instance of CIM_ManagedElement is the value of the property specified in the "Contained Class" column of Table 2 and the instance of CIM_ManagedElement referenced by the property specified in the "Container Class" column of Table 2 is in the scope of the instance of CIM_Role, where the scope is determined by recursively applying this algorithm.

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

7.1.2 CIM_Role.CommonName

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

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

7.1.3 Privileges of Authorized Role

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

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

The three types of CIM_Privilege instances are Denied Privileges, Granted Privileges, and Template Privileges (see sections 3.18, 3.19, and 3.22).

7.1.3.1 Granted Privileges and Denied Privileges

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

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

7.1.3.2 Cumulative Privileges for a Role

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

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

1) Select all the Granted Privileges (instances of CIM_Privilege with the PrivilegeGranted property set to TRUE) that are associated with the given CIM_Role instance through CIM_MemberOfCollection associations.

2) For each instance of Granted Privileges, select the CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats array properties.

3) For each element in the CIM_Privilege.Activities property array, select the value of the corresponding index of CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays,

   – Determine if the Cumulative Role Privilege’s CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays contain the combination of selected element values from step 3.

   – If not, add the combination of selected values to the appropriate array properties of Cumulative Role Privilege.

4) Select all the Denied Privileges (instances of CIM_Privilege with the PrivilegeGranted property set to FALSE) that are associated with the given CIM_Role instance through CIM_MemberOfCollection associations.

5) For each instance of Denied Privileges, select the CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats array properties.

6) For each element in the CIM_Privilege.Activities property array, select the value of the corresponding index of CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays,

   – Determine if the Cumulative Role Privilege’s CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays contain the combination of selected element values.

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

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

7.1.3.3 Cumulative Privileges for Multiple Roles

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

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

2) For each instance of Cumulative Role Privileges,

   – For each element in the CIM_Privilege.Activities property array, select the value of the corresponding index of CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays,

      1) Determine if the Cumulative Privilege’s CIM_Privilege.Activities, CIM_Privilege.ActivityQualifiers, and CIM_Privilege.QualifierFormats property arrays contain the combination of selected element values from step 1.

      2) If not, add the combination of selected values to the appropriate array properties of Cumulative Role Privilege.

7.1.3.4 Template Privileges

EXPERIMENTAL

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

EXPERIMENTAL

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

7.1.4 Static Authorized Role

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

7.2 Authorized Role Management

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

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

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

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

EXPERIMENTAL

- Creation of a CIM_Role instance and associated CIM_Privilege instances by using the CIM_RoleBasedAuthorizationService.CreateRole( ) method. See section 8.1 for requirement details.
- Deletion of a CIM_Role instance and associated CIM_Privilege instances by using the CIM_RoleBasedAuthorizationService.DeleteRole( ) method. See section 8.1.1 for requirement details.

EXPERIMENTAL

- Modification of a CIM_Role instance and associated CIM_Privilege instances by using the CIM_RoleBasedAuthorizationService.ModifyRole( ) method. See section 8.2.1 for requirement details.
- Modification of a CIM_Privilege instance by using the ModifyInstance operation. See section 8.13 for requirement details.

7.3 Authorized Role Membership of Security Principal

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

The *Simple Identity Management Profile* shall be implemented.

7.3.1 Roles Available to Principal

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

7.3.2 Managing Privileges through Role Assignment

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

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

7.3.3 Managing Privileges One to One for a Principal

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

Exactly one instance of CIM_ConcreteDependency shall be implemented as defined in section 10.2 that associates the CIM_Identity instance with a CIM_Role instance. At most one instance of CIM_Identity shall be associated with the CIM_Role instance through the CIM_MemberOfCollection association, if the CIM_Role instance is referenced by a CIM_ConcreteDependency association. The instance relationship through CIM_ConcreteDependency is used to indicate that the CIM_Role instance can be used for the single CIM_Identity instance with which it is associated.

7.4 Privilege Management Capability

This section provides requirements for identifying the Associated Privilege Management Capability for an instance of CIM_Privilege. Each instance of CIM_Privilege associated with a CIM_Role instance through CIM_MemberOfCollection association shall have the capabilities defined in the Associated Privilege Management Capability. CIM_Privilege may be optionally associated with CIM_RoleBasedAuthorizationService through CIM_ServiceAffectsElement association.

If there is an instance of CIM_ServiceAffectsElement associating the instance of CIM_Privilege with an instance of CIM_RoleBasedAuthorizationService, then the instance of CIM_RoleBasedManagementCapabilities associated with the instance of CIM_RoleBasedAuthorizationService shall be the Associated Privilege Management Capability.

If there is an instance of CIM_ServiceAffectsElement associating the instance of CIM_Privilege with an instance of CIM_RoleBasedAuthorizationService, the Associated Role Capability of instance(s) of CIM_Role associated with the instance of CIM_Privilege through CIM_MemberOfCollection association(s) shall be the Associated Privilege Management Capability.

7.4.1.1 Shared Privileges

If the CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported property is set to FALSE, the instance of CIM_Privilege shall be associated to only one instance of CIM_Role.

If the CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported property is set to TRUE, the instance of CIM_Privilege may be associated to one or more instances of CIM_Role.

7.4.1.2 Supported Activities

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

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

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

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

The ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties of the Associated Privilege Management Capability of the instance of CIM_Privilege shall represent the super set of supported activities, and the following rules apply:

- The CIM_Privilege.Activities property array shall contain a subset of elements of the ActivitiesSupported property array elements.
- The CIM_Privilege.ActivityQualifiers property array shall contain a subset of elements of the ActivityQualifiersSupported property array elements.

- The CIM_Privilege.QualifierFormats property array shall contain a subset of elements of the QualifierFormatsSupported property array elements.

8 Methods

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

EXPERIMENTAL

8.1 CIM_RoleBasedAuthorizationService.CreateRole( )

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

Upon the successful execution of the CreateRole( ) method:

- An instance of CIM_Role shall exist that is the exact replica of the embedded instance of CIM_Role of the RoleTemplate parameter except for the key properties.
- An instance of the CIM_OwningCollectionElement association shall associate the new CIM_Role instance and the scoping CIM_ComputerSystem instance referenced by the OwningSystem parameter.
- Instances of CIM_Privilege shall be associated with the newly created instance of CIM_Role through the CIM_MemberOfCollection association.
- The Cumulative Role Privilege of the newly associated instances of CIM_Privilege shall be equal to the Cumulative Role Privilege of the embedded instances of CIM_Privilege contained in the Privileges parameter.
- If the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of FALSE, the CIM_Privilege instances shall be associated only with the newly created CIM_Role instance and shall not be associated with any other instance of CIM_Role.
- If the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of TRUE, the CIM_Privilege instances shall be associated with the newly created CIM_Role instance and may be associated with any other instance of CIM_Role.
- Instances of CIM_RoleLimitedToTarget shall associate the newly created CIM_Role instance with the instances referenced by the RoleLimitedToTargets parameter.
- Instances of CIM_ServiceAffectsElement shall associate the new CIM_Role instance and the CIM_RoleBasedAuthorizationService instance.

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

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

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

No standard messages are defined for this method.

Table 3 – CIM_RoleBasedAuthorizationService.CreateRole( ) Method: Return Code Values

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

Table 4 – CIM_RoleBasedAuthorizationService.CreateRole( ) Method: Parameters

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

8.1.1 CIM_RoleBasedAuthorizationService.CreateRole( ) Conditional Support

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

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

8.2 CIM_RoleBasedAuthorizationService.DeleteRole( )

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

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

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

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

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

- All instances of the CIM_RoleLimitedToTarget association that reference the CIM_Role instance that is referenced by the Role parameter shall be deleted.
- If the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of FALSE, the implementation shall delete all the CIM_Privilege instances that are associated with the CIM_Role instance that is referenced by the Role parameter.
- If the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of TRUE, the implementation shall delete the CIM_Privilege instances that are only associated with the CIM_Role instance that is referenced by the Role parameter.
- All instances of the CIM_MemberOfCollection association that reference the CIM_Role instance that is referenced by the Role parameter shall be deleted.
- All instances of the CIM_OwningCollectionElement association that reference the CIM_Role instance that is referenced by the Role parameter shall be deleted.
- The instance of the CIM_ServiceAffectsElement association that references the CIM_Role instance that is referenced by the Role parameter and that references the CIM_RoleBasedAuthorizationService instance shall be deleted.

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

No standard messages are defined for this method.

Table 5 – CIM_RoleBasedAuthorizationService.DeleteRole( ) Method: Return Code Values

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

Table 6 – CIM_RoleBasedAuthorizationService.DeleteRole( ) Method: Parameters

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

8.2.1 CIM_RoleBasedAuthorizationService.DeleteRole( ) Conditional Support

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

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

EXPERIMENTAL

8.3 CIM_RoleBasedAuthorizationService.ModifyRole( )

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

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

- If the Privileges parameter is Null, the instances of CIM_Privilege that are associated with the Modified Role shall not be modified (see section 3.20).
- If the Privileges parameter is not Null and instances of CIM_Privilege are associated with the Modified Role through the CIM_MemberOfCollection association, the Cumulative Role Privilege of the associated instances of CIM_Privilege shall be equal to the Cumulative Role Privilege of the embedded instances of CIM_Privilege that are contained in the Privileges parameter.
- If the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of FALSE, the CIM_Privilege instances shall be associated only with the Modified Role and shall not be associated with any other instance of CIM_Role.
- If the SharedPrivilegeSupported property of the CIM_RoleBasedManagementCapabilities instance that is associated with the CIM_RoleBasedAuthorizationService instance has a value of TRUE, the CIM_Privilege instances shall be associated with the Modified Role and may be associated with any other instance of CIM_Role.
- An instance of CIM_RoleLimitedToTarget shall reference the Modified Role and an instance of CIM_ManagedElement only if a reference to the CIM_ManagedElement was contained in the RoleLimitedToTargets parameter.

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

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

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

No standard messages are defined for this method.

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

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

Table 8 – CIM_RoleBasedAuthorizationService.ModifyRole( ) Method: Parameters

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

8.3.1 CIM_RoleBasedAuthorizationService.ModifyRole( ) Conditional Support

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

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

8.4 CIM_RoleBasedAuthorizationService.AssignRoles( )

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

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

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

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

No standard messages are defined for this method.

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

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

Table 10 – CIM_RoleBasedAuthorizationService.AssignRoles( ) Method: Parameters

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

8.4.1 CIM_RoleBasedAuthorizationService.AssignRoles( ) Conditional Support

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

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

8.5 CIM_RoleBasedAuthorizationService.ShowAccess( )

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

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

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

If the CIM_Identity instance identified by the Subject parameter is not associated with an instance of CIM_AccountManagementService instance through the CIM_ServiceAffectsElement association, where the CIM_AccountManagementService is associated through the CIM_ServiceServiceDependency association with the instance of CIM_RoleBasedAuthorizationService upon which the method was invoked, the method shall return the value 2 (Failed).

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

- the set of instances of CIM_Role are those instances such that the instance of CIM_Identity specified by the Subject parameter is a member of the CIM_Role instance as defined in section 7.3.2
- the instance of CIM_ManagedElement specified by the Target parameter is in the scope of the CIM_Role instance as defined in section 7.1.1.1
- the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService through the CIM_ServiceAffectsElement association

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

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

No standard messages are defined for this method.

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

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

Table 12 – CIM_RoleBasedAuthorizationService.ShowAccess( ) Method: Parameters

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

8.5.1 CIM_RoleBasedAuthorizationService.ShowAccess( ) Conditional Support

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

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

8.6 CIM_RoleBasedAuthorizationService.ShowRoles( )

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

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

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

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

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

- the Roles parameter shall contain an embedded instance of CIM_Role for each instance of CIM_Role such that the instance of CIM_Identity specified by the Subject parameter is a member of the CIM_Role instance as defined in section 7.3.2
- the instance of CIM_ManagedElement specified by the Target parameter is in the scope of the CIM_Role instance as defined in section 7.1.1.1
- the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService through the CIM_ServiceAffectsElement association

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

- the Roles parameter shall contain an embedded instance of CIM_Role for each of instance of CIM_Role such that the instance of CIM_Identity specified by the Subject parameter is a member of the CIM_Role instance as defined in section 7.3.2
- the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService through the CIM_ServiceAffectsElement association

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

- the Roles parameter shall contain an embedded instance of CIM_Role for each of instance of CIM_Role such that the instance of CIM_ManagedElement specified by the Target parameter is in the scope of the CIM_Role instance as defined in section 7.1.1.1
- the instance of CIM_Role is associated with the instance of CIM_RoleBasedAuthorizationService through the CIM_ServiceAffectsElement association

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

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

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

No standard messages are defined for this method.

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

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

Table 14 – CIM_RoleBasedAuthorizationService.ShowRoles( ) Method: Parameters

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

8.6.1 CIM_RoleBasedAuthorizationService.ShowRoles( ) Conditional Support

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

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

8.7 Profile Conventions for Operations

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

The default list of operations is as follows:

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

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

8.8 CIM_ConcreteDependency

Table 15 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 15 – Operations: CIM_ConcreteDependency

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

8.9 CIM_ElementCapabilities

Table 16 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 16 – Operations: CIM_ElementCapabilities

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

8.10 CIM_HostedService

Table 17 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 17 – Operations: CIM_HostedService

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

8.11 CIM_MemberOfCollection

Table 18 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 18 – Operations: CIM_MemberOfCollection

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

8.12 CIM_OwningCollectionElement

Table 19 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 19 – Operations: CIM_OwningCollectionElement

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

8.13 CIM_Privilege

Table 20 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 20 – Operations: CIM_Privilege

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.13.1.	None

8.13.1 CIM_Privilege—ModifyInstance

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

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

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

8.14 CIM_RoleBasedManagementCapabilities

All operations in the default list in section 8.7 are supported as described by DSP0200 version 1.2.

8.15 CIM_Role

All operations in the default list in section 8.7 are supported as described by DSP0200 version 1.2.

8.16 CIM_RoleBasedAuthorizationService

All operations in the default list in section 8.7 are supported as described by DSP0200 version 1.2.

8.17 CIM_RoleLimitedToTarget

Table 21 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 21 – Operations: CIM_RoleLimitedToTarget

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

8.18 CIM_ServiceAffectsElement

Table 22 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 22 – Operations: CIM_ServiceAffectsElement

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

8.19 CIM_ServiceServiceDependency

Table 23 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 23 – Operations: CIM_ServiceServiceDependency

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

9 Use Cases

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

9.1 Profile Registration

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

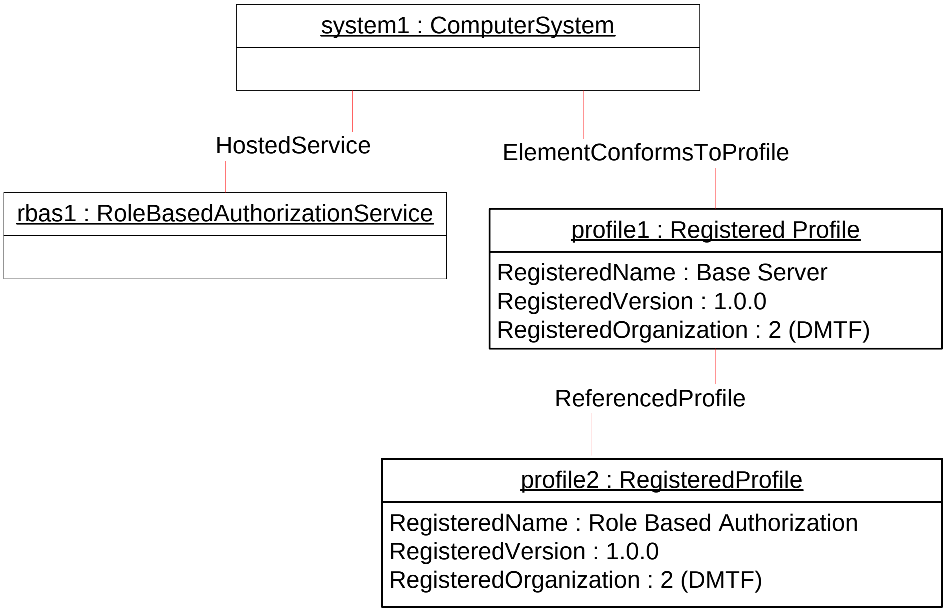

system1 : ComputerSystem
HostedService
ElementConformsToProfile
rbas1 : RoleBasedAuthorizationService
profile1 : Registered Profile
RegisteredName : Base Server
RegisteredVersion : 1.0.0
RegisteredOrganization : 2 (DMTF)
ReferencedProfile
profile2 : RegisteredProfile
RegisteredName : Role Based Authorization
RegisteredVersion : 1.0.0
RegisteredOrganization : 2 (DMTF)

Figure 2 – Profile Registration

9.2 Minimal Instantiation of the Profile

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

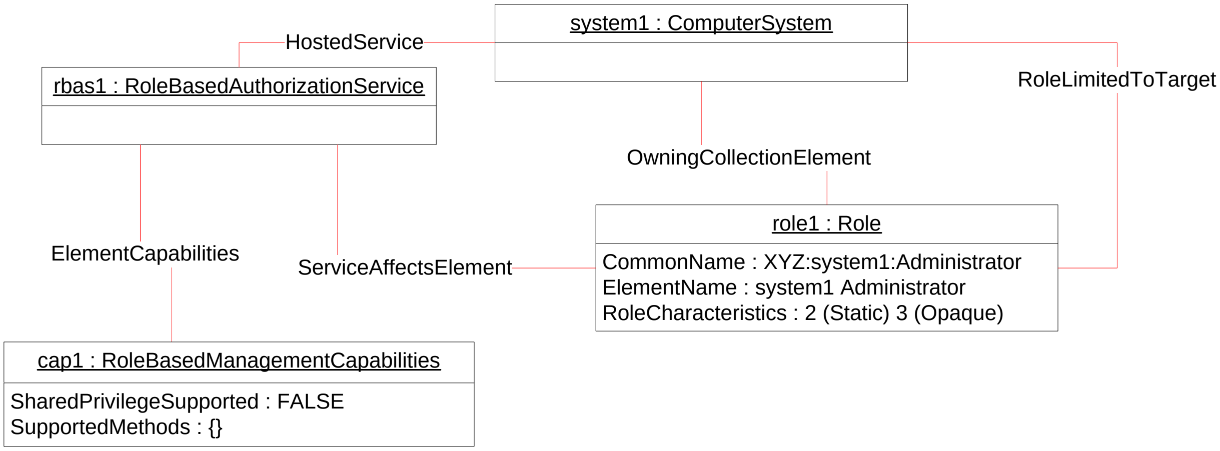

system1 : ComputerSystem
HostedService
RoleLimitedToTarget
rbas1 : RoleBasedAuthorizationService
OwningCollectionElement
role1 : Role
CommonName : XYZ:system1:Administrator
ElementName : system1 Administrator
RoleCharacteristics : 2 (Static) 3 (Opaque)
ElementCapabilities
ServiceAffectsElement
cap1 : RoleBasedManagementCapabilities
SharedPrivilegeSupported : FALSE
SupportedMethods : {}

Figure 3 – Minimal Instantiation

9.3 Evaluating Scope and Privileges

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

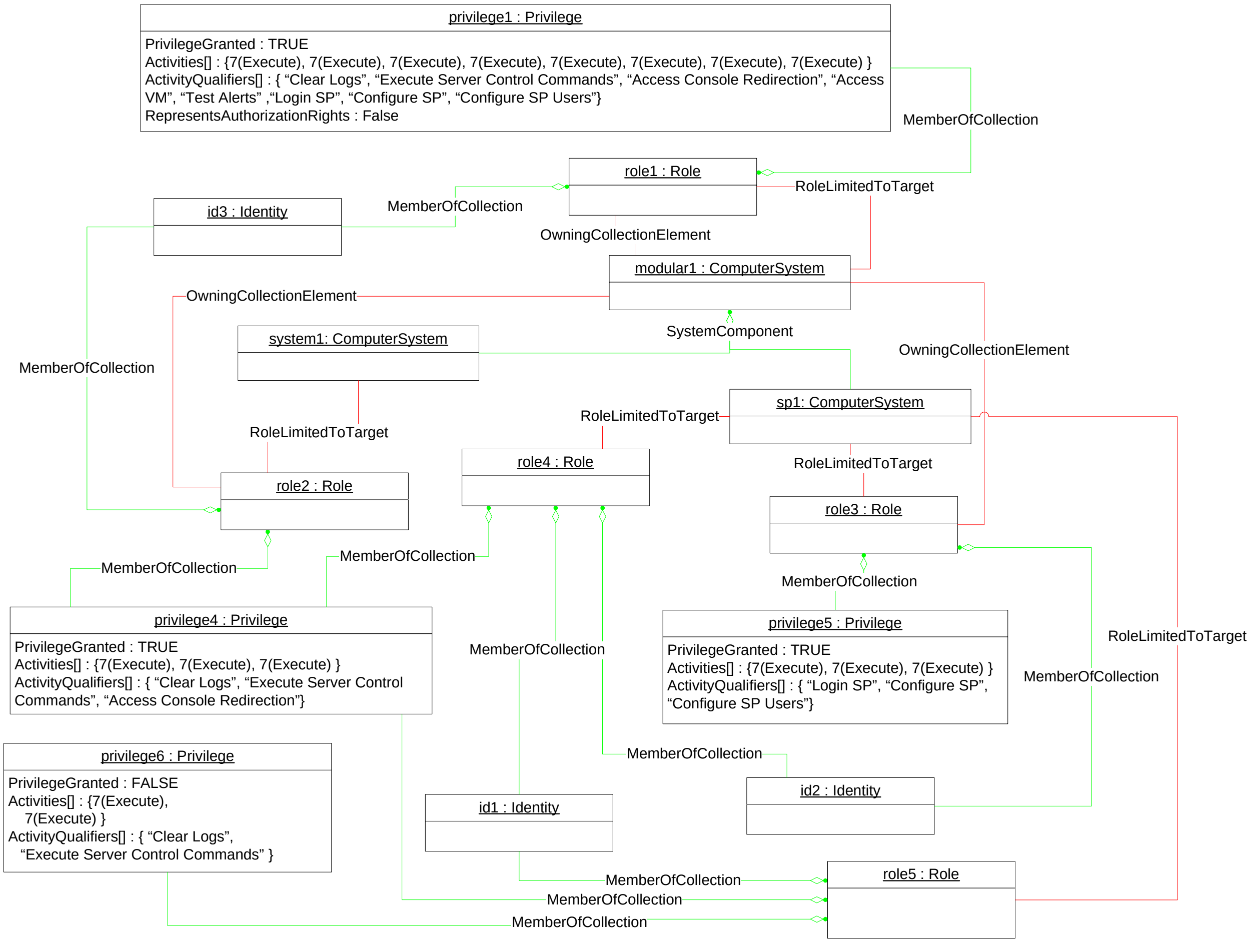

privilege1 : Privilege
PrivilegeGranted : TRUE
Activities[] : {7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute) }
ActivityQualifiers[] : { "Clear Logs", "Execute Server Control Commands", "Access Console Redirection", "Access VM", "Test Alerts" ,"Login SP", "Configure SP", "Configure SP Users"}
RepresentsAuthorizationRights : False
MemberOfCollection
role1 : Role
RoleLimitedToTarget
id3 : Identity
MemberOfCollection
OwningCollectionElement
modular1 : ComputerSystem
OwningCollectionElement
system1: ComputerSystem
SystemComponent
OwningCollectionElement
MemberOfCollection
sp1: ComputerSystem
RoleLimitedToTarget
RoleLimitedToTarget
role4 : Role
RoleLimitedToTarget
role2 : Role
role3 : Role
MemberOfCollection
MemberOfCollection
MemberOfCollection
privilege4 : Privilege
PrivilegeGranted : TRUE
Activities[] : {7(Execute), 7(Execute), 7(Execute) }
ActivityQualifiers[] : { "Clear Logs", "Execute Server Control Commands", "Access Console Redirection"}
MemberOfCollection
privilege5 : Privilege
PrivilegeGranted : TRUE
Activities[] : {7(Execute), 7(Execute), 7(Execute) }
ActivityQualifiers[] : { "Login SP", "Configure SP", "Configure SP Users"}
RoleLimitedToTarget
MemberOfCollection
privilege6 : Privilege
PrivilegeGranted : FALSE
Activities[] : {7(Execute), 7(Execute) }
ActivityQualifiers[] : { "Clear Logs", "Execute Server Control Commands" }
MemberOfCollection
id2 : Identity
id1 : Identity
role5 : Role
MemberOfCollection
MemberOfCollection
MemberOfCollection

Figure 4 – Cumulative Role Privilege Example

9.3.1 CIM_RoleBasedManagementService.ShowRoles( )

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

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

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

9.3.2 CIM_RoleBasedManagementService.ShowAccess( )

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

9.3.2.1 Example: CIM_RoleBasedManagementService.ShowAccess( )

Subject = id1

Target = sp1

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

CIM_Privilege.ActivityQualifiers = { “Clear Logs”, “Execute Server Control Commands”, “Access Console Redirection” }

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

9.3.2.2 Example: CIM_RoleBasedManagementService.ShowAccess( )

Subject = id3

Target = modular1

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

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

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

9.3.2.3 Example: CIM_RoleBasedManagementService.ShowAccess( )

Subject = id3

Target = system1

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

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

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

9.3.2.4 Example: CIM_RoleBasedManagementService.ShowAccess( )

Subject = id3

Target = sp1

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

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

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

9.3.2.5 Example: CIM_RoleBasedManagementService.ShowAccess( )

Subject = id2

Target = sp1

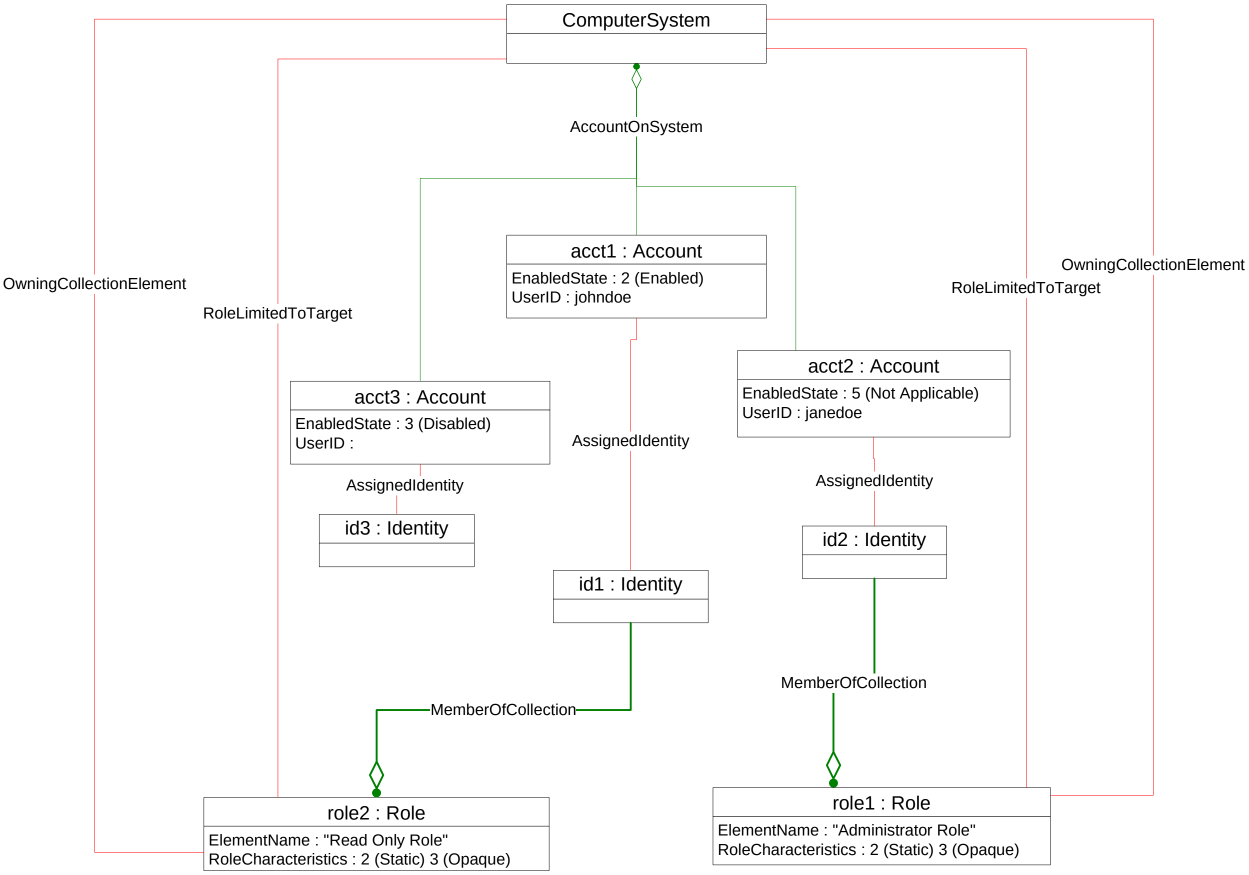
CIM_Privilege.Activities = { 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute) }

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

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

9.4 Scope of the Role and Privileges for a Managed Element

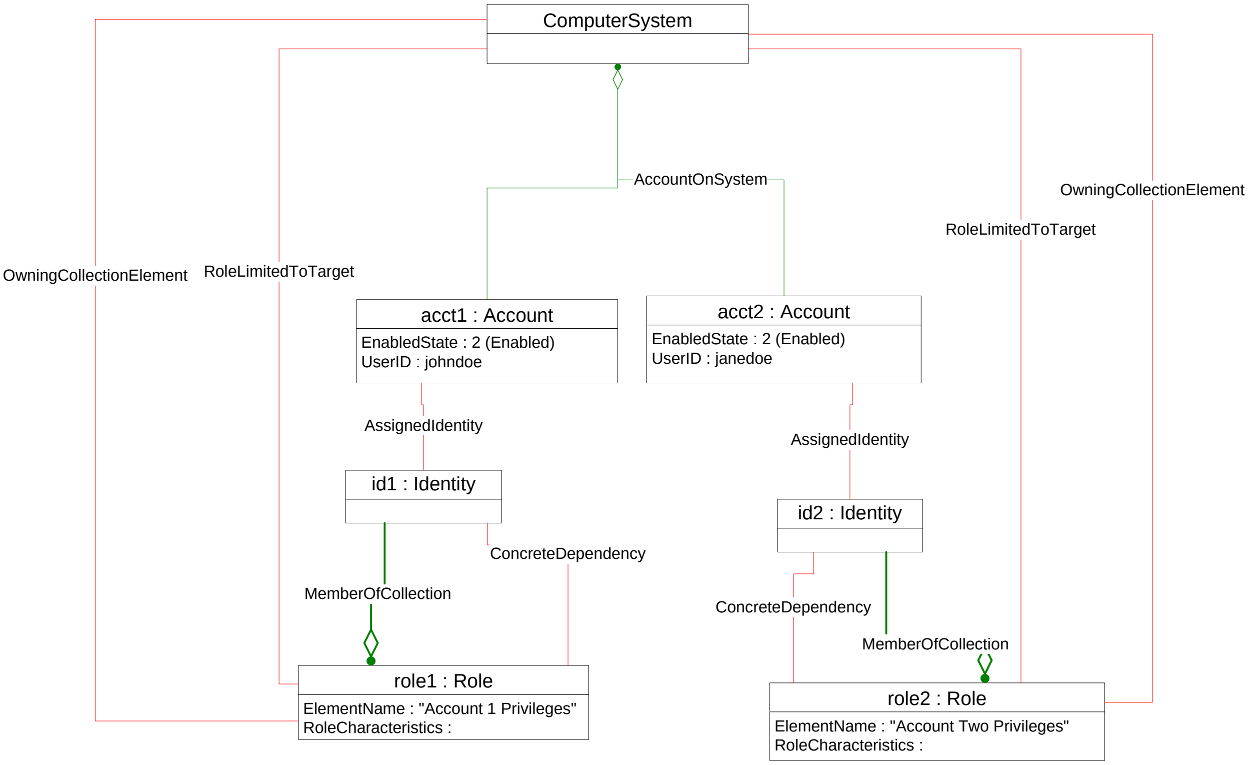
Figure 5 shows a system that has three local accounts and uses role membership to manage the privileges for a user account. This system has three local accounts: acct1, acct2, and acct3. acct1 currently has privileges of role1, and acct2 currently has the privileges of role2. acct3 does not have any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



1024

1025 **Figure 5 – Scope of the Roles**

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



1031

1032 **Figure 6 – Fixed Accounts with Role Membership Privilege Management**

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

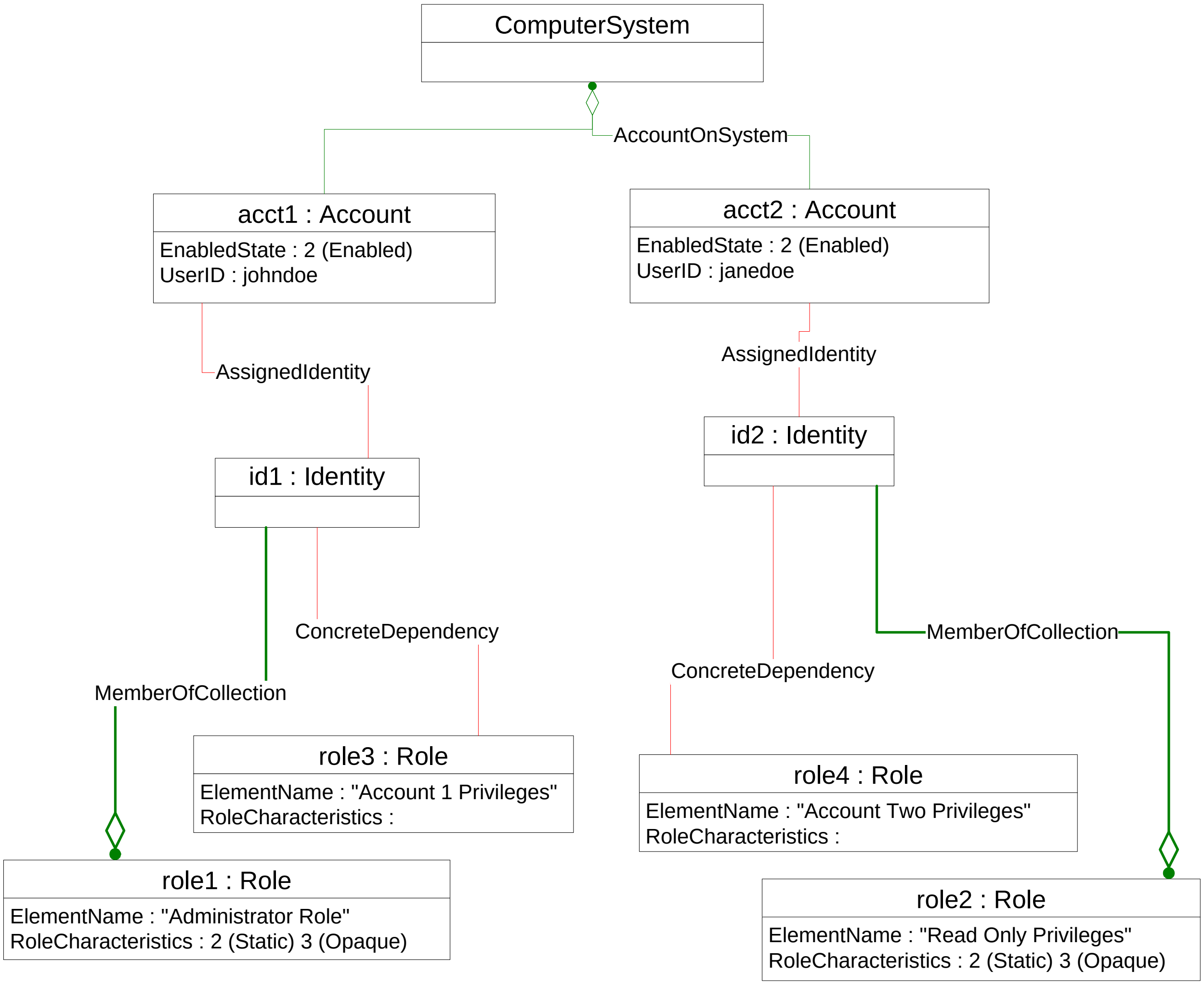


Figure 7 – Fixed Accounts with Individual Account Privilege Management

9.5 Service Processor Roles Use Cases

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

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

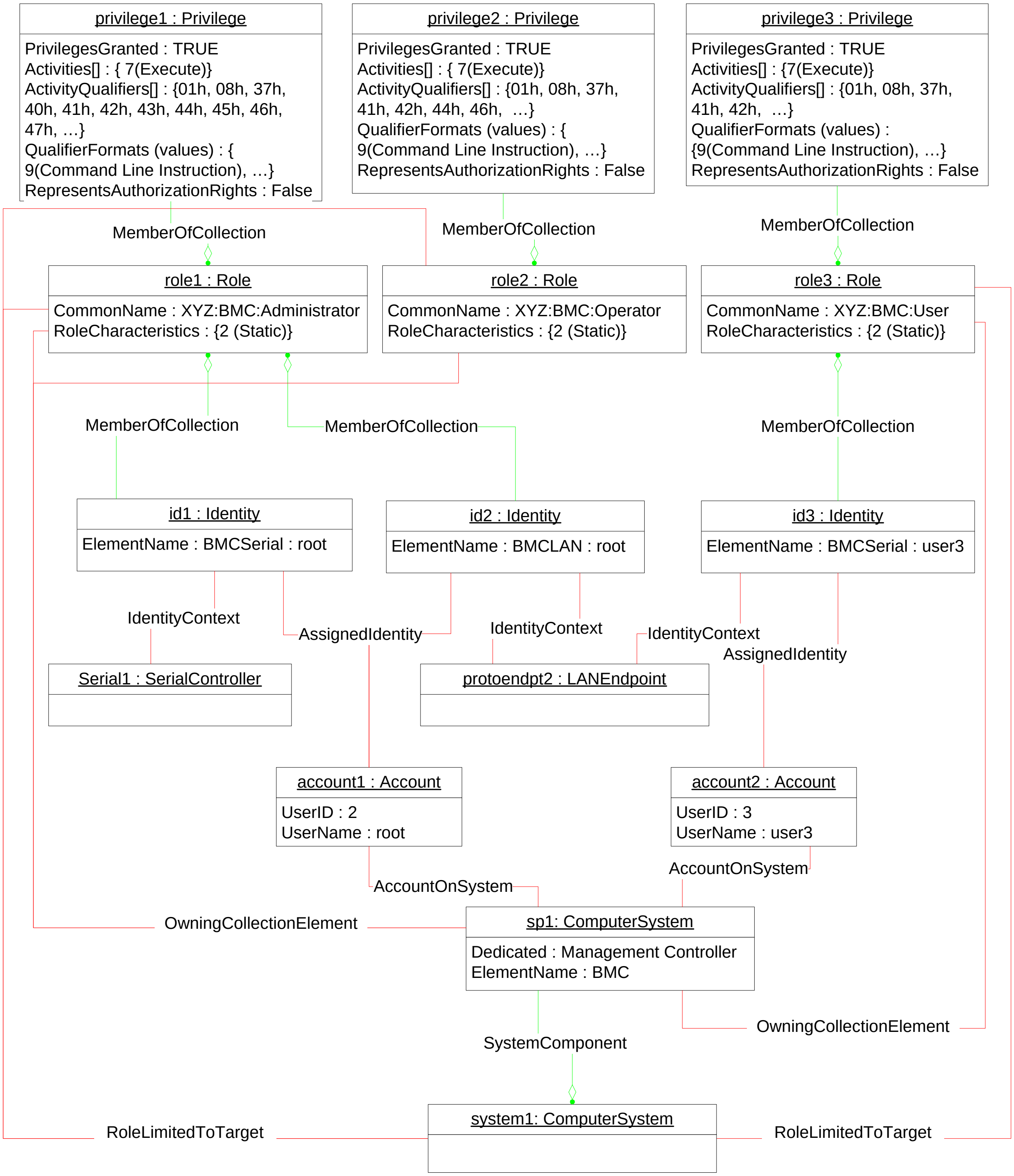

privilege1 : Privilege
PrivilegesGranted : TRUE
Activities[] : { 7(Execute)}
ActivityQualifiers[] : {01h, 08h, 37h, 40h, 41h, 42h, 43h, 44h, 45h, 46h, 47h, …}
QualifierFormats (values) : { 9(Command Line Instruction), …}
RepresentsAuthorizationRights : False
privilege2 : Privilege
PrivilegesGranted : TRUE
Activities[] : { 7(Execute)}
ActivityQualifiers[] : {01h, 08h, 37h, 41h, 42h, 44h, 46h, …}
QualifierFormats (values) : { 9(Command Line Instruction), …}
RepresentsAuthorizationRights : False
privilege3 : Privilege
PrivilegesGranted : TRUE
Activities[] : {7(Execute)}
ActivityQualifiers[] : {01h, 08h, 37h, 41h, 42h, …}
QualifierFormats (values) : {9(Command Line Instruction), …}
RepresentsAuthorizationRights : False
MemberOfCollection
MemberOfCollection
MemberOfCollection
role1 : Role
CommonName : XYZ:BMC:Administrator
RoleCharacteristics : {2 (Static)}
role2 : Role
CommonName : XYZ:BMC:Operator
RoleCharacteristics : {2 (Static)}
role3 : Role
CommonName : XYZ:BMC:User
RoleCharacteristics : {2 (Static)}
MemberOfCollection
MemberOfCollection
MemberOfCollection
id1 : Identity
ElementName : BMCSerial : root
id2 : Identity
ElementName : BMCLAN : root
id3 : Identity
ElementName : BMCSerial : user3
IdentityContext
AssignedIdentity
IdentityContext
IdentityContext
AssignedIdentity
Serial1 : SerialController
protoendpt2 : LANEndpoint
account1 : Account
UserID : 2
UserName : root
account2 : Account
UserID : 3
UserName : user3
AccountOnSystem
AccountOnSystem
OwningCollectionElement
sp1: ComputerSystem
Dedicated : Management Controller
ElementName : BMC
OwningCollectionElement
SystemComponent
system1: ComputerSystem
RoleLimitedToTarget
RoleLimitedToTarget

Figure 8 – IPMI Service Processor with Role Management

EXPERIMENTAL

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

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

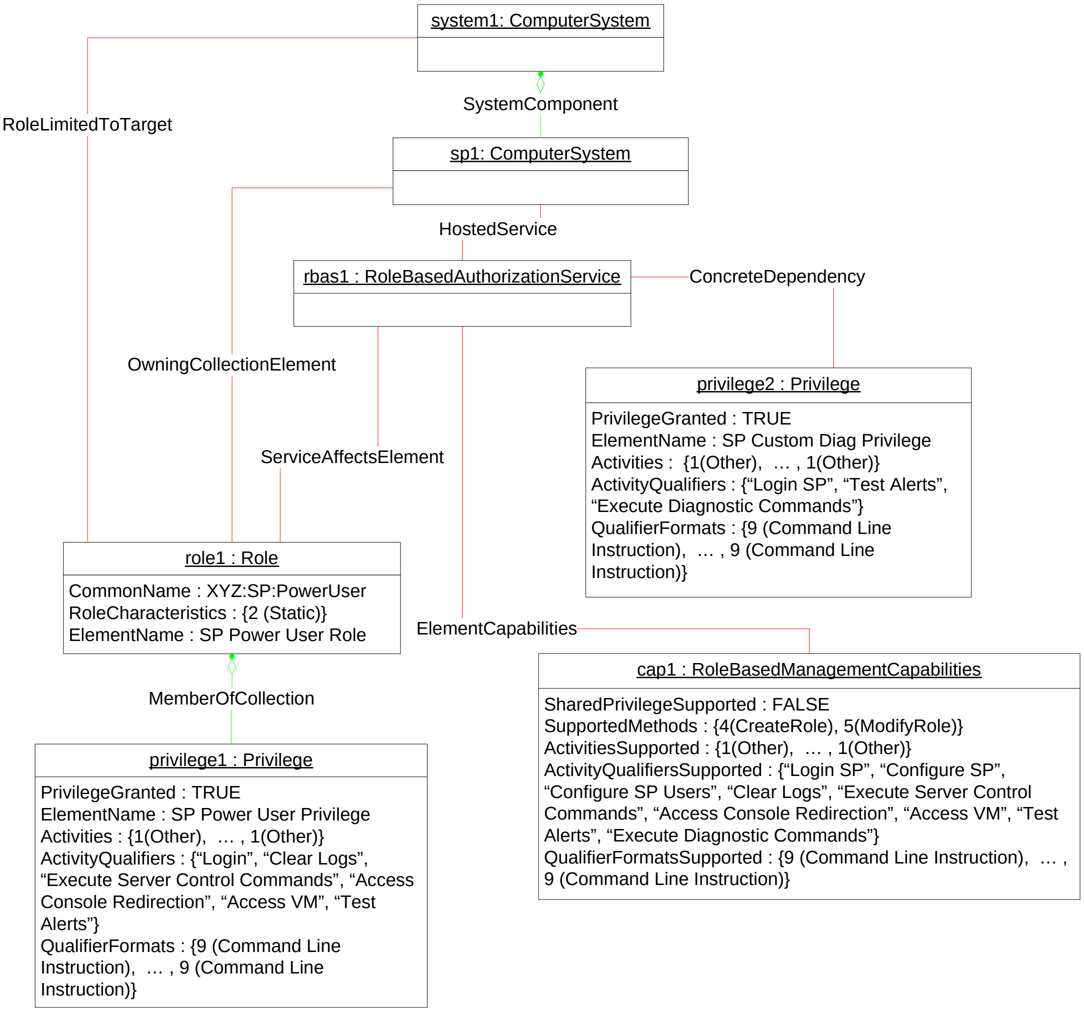

system1: ComputerSystem
SystemComponent
RoleLimitedToTarget
sp1: ComputerSystem
HostedService
rbas1 : RoleBasedAuthorizationService
ConcreteDependency
OwningCollectionElement
privilege2 : Privilege
PrivilegeGranted : TRUE
ElementName : SP Custom Diag Privilege
Activities : {1(Other), … , 1(Other)}
ActivityQualifiers : {"Login SP", "Test Alerts", "Execute Diagnostic Commands"}
QualifierFormats : {9 (Command Line Instruction), … , 9 (Command Line Instruction)}
ServiceAffectsElement
role1 : Role
CommonName : XYZ:SP:PowerUser
RoleCharacteristics : {2 (Static)}
ElementName : SP Power User Role
ElementCapabilities
cap1 : RoleBasedManagementCapabilities
SharedPrivilegeSupported : FALSE
SupportedMethods : {4(CreateRole), 5(ModifyRole)}
ActivitiesSupported : {1(Other), … , 1(Other)}
ActivityQualifiersSupported : {"Login SP", "Configure SP", "Configure SP Users", "Clear Logs", "Execute Server Control Commands", "Access Console Redirection", "Access VM", "Test Alerts", "Execute Diagnostic Commands"}
QualifierFormatsSupported : {9 (Command Line Instruction), … , 9 (Command Line Instruction)}
MemberOfCollection
privilege1 : Privilege
PrivilegeGranted : TRUE
ElementName : SP Power User Privilege
Activities : {1(Other), … , 1(Other)}
ActivityQualifiers : {"Login", "Clear Logs", "Execute Server Control Commands", "Access Console Redirection", "Access VM", "Test Alerts"}
QualifierFormats : {9 (Command Line Instruction), … , 9 (Command Line Instruction)}

Figure 9 – IPMI Service Processor with Role Management

EXPERIMENTAL

9.6 Determine the Roles Managed by a Service

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

1) Find the instance of CIM_RoleBasedManagementCapabilities that is associated with the target instance through an instance of CIM_ElementCapabilities.

2) If the CIM_RoleBasedManagementCapabilities.SupportedMethods property contains the value 7 (ShowRoles), invoke the CIM_RoleBasedAuthorizationService.ShowRoles( ) method, specifying Null for the Subject and Target parameters.

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

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

9.7 Determine Candidate Roles for a Security Principal

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

1) Find the instance of CIM_AccountManagementService that is associated with the CIM_Identity instance through the CIM_ServiceAffectsElement association.

2) Find the instances of CIM_RoleBasedAuthorizationService that are associated with the CIM_AccountManagementService through the CIM_ServiceServiceDependency association.

3) For each instance of CIM_RoleBasedAuthorizationService, use the steps in section 9.6 to find the instances of CIM_Role that are managed by the service.

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

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

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

1) Find the instance of CIM_AccountManagementService that is associated with the CIM_Identity instance through the CIM_ServiceAffectsElement association.

2) Find the instances of CIM_RoleBasedAuthorizationService that are associated with the CIM_AccountManagementService through the CIM_ServiceServiceDependency association.

3) For each instance of CIM_RoleBasedAuthorizationService, find the instance of CIM_RoleBasedManagementCapabilities that is associated through the CIM_ElementCapabilities association.

4) If the CIM_RoleBasedManagementCapabilities.SupportedMethods property contains the value 7 (ShowRoles),

   1) Invoke the CIM_RoleBasedAuthorizationService.ShowRoles( ) method, specifying a reference to the CIM_Identity instance as the value of the Subject parameter and Null for the Target parameter.

   2) Upon successful completion, the Roles parameter will contain an embedded instance of CIM_Role for each CIM_Role instance managed by the service, and the Privileges parameter will contain an instance of Cumulative Privilege for each returned instance of CIM_Role.

5) Else, if the CIM_RoleBasedManagementCapabilities.SupportedMethods property does not contain the value 7 (ShowRoles), find all of the instances of CIM_Role that are associated with the CIM_Identity instance through the CIM_MemberOfCollection association.

9.9 Determine the Roles that Scope a Managed Element

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

1) Enumerate all the instances of CIM_RoleBasedAuthorizationService.

2) For each instrumented instance of CIM_RoleBasedAuthorizationService, find the instance of CIM_RoleBasedManagementCapabilities that is associated through the CIM_ElementCapabilities association.

3) If the CIM_RoleBasedManagementCapabilities.SupportedMethods property contains the value 7 (ShowRoles), invoke the ShowRoles( ) method, specifying Null for the Subject parameter and a reference to the CIM_ManagedElement instance as the value of the Target parameter.

4) Else, use the traversal algorithm described in section 7.1.1.1.

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

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

1) Find the instance of CIM_AccountManagementService that is associated with the CIM_Identity instance through the CIM_ServiceAffectsElement association.

2) Find the instances of CIM_RoleBasedAuthorizationService that are associated with the CIM_AccountManagementService through the CIM_ServiceServiceDependency association.

3) For each instance of CIM_RoleBasedAuthorizationService, find the instance of CIM_RoleBasedManagementCapabilities that is associated through the CIM_ElementCapabilities association.

4) If the CIM_RoleBasedManagementCapabilities.SupportedMethods property contains the value 1 (ShowAccess), invoke the CIM_RoleBasedAuthorizationService.ShowAccess( ) method, specifying a reference to the CIM_Identity instance as the value of the Subject parameter and a reference to the instance of CIM_ManagedElement for the Target parameter.

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

5) Else, construct the Cumulative Privilege as defined in section 7.1.3.3, where the set of instances of CIM_Role are those instances such that the given instance of CIM_Identity is a member of the CIM_Role instance as defined in section 7.3.2, and the given instance of CIM_ManagedElement specified by the Target parameter is in the scope of the CIM_Role instance as defined in section 7.1.1.1.

9.11 Modify a Single Privilege of an Existing Role

For a given instance of CIM_Role that represents an existing role, a client can modify a single privilege of the role as follows:

1) If the RoleCharacteristics property of the selected instance of CIM_Role does not have the value 2 (Static), then select the Associated Role Management Capability of the selected CIM_Role instance,
    1) If the SupportedMethods property of the Associated Privilege Management Capability of the selected CIM_Privilege instance has a value of 8 (ModifyPrivilege),
        1) Execute the ModifyInstance operation on the selected instances of CIM_Privilege, modifying the privilege accordingly.
    2) Else, the privileges cannot be modified.
2) Else, the role is static and its privileges cannot be modified.

EXPERIMENTAL

9.12 Create a New Role

For a given instance of CIM_RoleBasedAuthorizationService, a client can create a new role as follows:

1) Find the CIM_RoleBasedManagementCapabilities instance associated to the given instance of CIM_RoleBasedAuthorizationService.
2) If the SupportedMethods property of the CIM_RoleBasedManagementCapabilities instance has a value of 4 (CreateRole),
    1) Construct the parameters for the CIM_RoleBasedAuthorizationService.CreateRole( ) method in the following way:
        - RoleTemplate: Construct the desired embedded instance of CIM_Role.
        - OwningSystem: Construct the CIM reference to the instance of CIM_ComputerSystem that will be the Scoping Instance of the newly created instance of CIM_Role.
        - Privileges: Construct the embedded instance of CIM_Privilege based on the ActivitiesSupported, ActivityQualifiersSupported, and QualifierFormatsSupported properties of the selected instance of CIM_RoleBasedManagementCapabilities, or based on the Template Privilege associated with the CIM_RoleBasedAuthorizationService instance.
        - RoleLimitedToTargets: Construct the CIM reference to the instance of subclass of CIM_ManagedElement which will be the Root Instance of the new instance of CIM_Role.
    2) Execute the CIM_RoleBasedAuthorizationService.CreateRole( ) method with the preceding parameters.
3) Else, the given instance of CIM_RoleBasedAuthorizationService does not support the creation of new role and this use case is not supported.

EXPERIMENTAL

9.13 Determine Whether Privilege Management Is Supported for a Principal

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

1) Starting at the instance of CIM_Identity that represents the security principal, find the instances of CIM_AccountManagementService that are associated through the CIM_ServiceAffectsElement association.

2) For each instance of CIM_AccountManagementService, determine if at least one instance of CIM_RoleBasedAuthorizationService is associated through the CIM_ServiceServiceDependency association.

3) If at least one instance of CIM_RoleBasedAuthorizationService is associated with at least one instance of CIM_AccountManagementService, privilege management is supported for the security principal.

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

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

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

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

9.15 Assign Custom Privileges to an Identity

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

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

   If privileges are not managed through one-to-one correspondence, it is necessary to create a custom role that has the desired privileges. See section 9.12 for information about how to create a role with the desired privileges.

2) If privileges are managed through one-to-one correspondence, find the instance of CIM_Identity that is associated with the CIM_Account instance.

3) Find the instance of CIM_Role that is associated with the CIM_Identity instance through an instance of CIM_ConcreteDependency.

4) If the CIM_Identity instance is not already associated with the instance of CIM_Role from step 3) through an instance of CIM_MemberOfCollection, use CreateInstance to create an instance of CIM_MemberOfCollection that associates the CIM_Identity instance with the CIM_Role instance.

5) If the CIM_Identity is associated with the instance of CIM_Role other than that from step 3) through an instance of CIM_MemberOfCollection, use DeleteInstance to delete the instance of CIM_MemberOfCollection that associates the CIM_Identity instance with the CIM_Role instance.

6) Perform role modification on the instance of CIM_Role from step 3) as specified in section 9.6.

10 CIM Elements

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

Table 24 – CIM Elements: Role Based Authorization Profile

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

10.1 CIM_ConcreteDependency (Privilege)

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

Table 25 – Class: CIM_ConcreteDependency (Privilege)

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

10.2 CIM_ConcreteDependency (Role)

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

Table 26 – Class: CIM_ConcreteDependency (Role)

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

10.3 CIM_ElementCapabilities

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

Table 27 – Class: CIM_ElementCapabilities

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

10.4 CIM_HostedService

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

Table 28 – Class: CIM_HostedService

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

10.5 CIM_MemberOfCollection (Privilege)

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

Table 29 – Class: CIM_MemberOfCollection (Privilege)

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

10.6 CIM_MemberOfCollection (Identity)

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

Table 30 – Class: CIM_MemberOfCollection (Identity)

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

10.7 CIM_OwningCollectionElement

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

Table 31 – Class: CIM_OwningCollectionElement

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

10.8 CIM_Privilege

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

Table 32 – Class: CIM_Privilege

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

10.9 CIM_RoleBasedManagementCapabilities

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

Table 33 – Class: CIM_RoleBasedManagementCapabilities

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

10.10 CIM_RegisteredProfile

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

Table 34 – Class: CIM_RegisteredProfile

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

10.11 CIM_Role

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

Table 35 – Class: CIM_Role

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

10.12 CIM_RoleBasedAuthorizationService

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

Table 36 – Class: CIM_RoleBasedAuthorizationService

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

10.13 CIM_RoleLimitedToTarget

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

Table 37 – Class: CIM_RoleLimitedToTarget

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

10.14 CIM_ServiceAffectsElement – CIM_Role

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

Table 38 – Class: CIM_ServiceAffectsElement

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

10.15 CIM_ServiceAffectsElement – CIM_Privilege

If the instance of CIM_Privilege is associated with instances of CIM_Role which are in turn associated with different instances of CIM_RoleBasedAuthorizationService, CIM_ServiceAffectsElement associating CIM_Privilege with a CIM_RoleBasedAuthorizationService instance shall be implemented.

CIM_ServiceAffectsElement is used to associate an instance of CIM_RoleBasedAuthorizationService with an instance of CIM_Privilege that represents a privilege. Table 39 contains the requirements for elements of this class.

Table 39 – Class: CIM_ServiceAffectsElement

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

10.16 CIM_ServiceServiceDependency

CIM_ServiceServiceDependency is used to associate an instance of CIM_RoleBasedAuthorizationService with an instance of CIM_AccounManagementService representing that the identities of the CIM_AccountManagmentService instance could be members of roles of the associated CIM_RoleBasedAuthorizationService instance. Table 40 contains the requirements for elements of this class.

1303 **Table 40 – Class: CIM_ServiceServiceDependency**

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

ANNEX A
(informative)

Change Log

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

ANNEX B
(informative)

Acknowledgements

The authors wish to acknowledge the following people.

Editors:

- Khachatur Papanyan – Dell
- Aaron Merkin – IBM

Contributors:

- Murali Rajagopal – Broadcom
- Hemal Shah – Broadcom
- Jon Hass – Dell
- Khachatur Papanyan – Dell
- George Ericson – EMC
- Christina Shaw – HP
- Aaron Merkin – IBM
- David Hines – Intel