Document Number: DSP1034

Date: 2008-07-23

Version: 1.0.0

Simple Identity Management Profile

Document Type: Specification

Document Status: Final Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

The *Simple Identity Management Profile* (DSP1034) was prepared by the WBEM Infrastructures and Protocols Working Group of the DMTF.

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

Introduction

The information in this specification should be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to represent and manage an Account and its Security Principal that is 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.

Simple Identity Management Profile

1 Scope

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

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*

ANSI T1.276-2003, *Operations, Administration, Maintenance, and Provisioning Security Requirements for the Public Telecommunications Network: A Baseline of Security Requirements for the Management Plane*

2.2 References under Development

DMTF DSP1033, *Profile Registration Profile*

DMTF DSP1039, *Role Based Authorization Profile*

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, DSP1001, and T1.276-2003 also apply.

3.1
account identity
The security principal that represents an authenticated Account.

3.2
can
used for statements of possibility and capability, whether material, physical, or causal

3.3
cannot
used for statements of possibility and capability, whether material, physical, or causal

3.4
conditional
indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met

3.5
mandatory
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.6
may
indicates a course of action permissible within the limits of the document

3.7
need not
indicates a course of action permissible within the limits of the document

3.8
optional
indicates a course of action permissible within the limits of the document

3.9
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

3.10
shall
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.11
shall not
indicates requirements to be followed in order to conform to the document and from which no deviation is permitted

3.12
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

3.13
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

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

3.15
authentication
the process of verifying the credentials provided by an entity for the purpose of resolving to a security principal

3.16
first-party authentication
authentication that is performed using services that execute local to the relying party

3.17
principal
an entity that can be positively identified and verified through an authentication mechanism

3.18
third-party authentication
authentication that is performed using services that execute remote to the relying party

4 Symbols and Abbreviated Terms

Experimental Maturity Level

Some of the content considered for inclusion in *Simple Identity Management 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: *Simple Identity Management*

Version: 1.0.0

Organization: DMTF

CIM schema version: 2.19.0

Central Class: CIM_AccountManagementService

297 **Scoping Class:** CIM_ComputerSystem

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

300 CIM_AccountManagementService shall be the Central Class of this profile. The instance of
301 CIM_AccountManagementService shall be the Central Instance of this profile. CIM_ComputerSystem
302 shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem with which the Central
303 Instance is associated through an instance of CIM_HostedService shall be the Scoping Instance of this
304 profile.

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

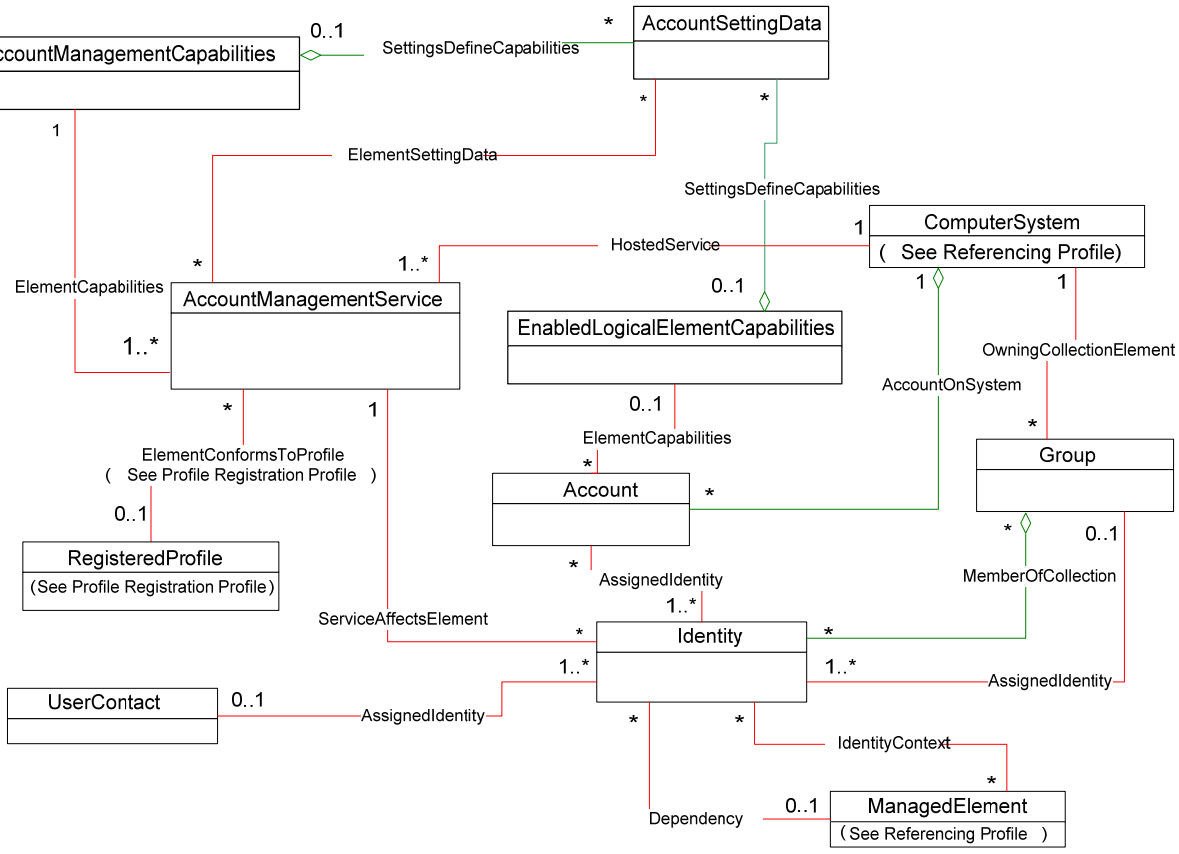
306 **Table 1 – Referenced Profiles**

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0.0	Mandatory	

307 # 6 Description

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

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



313

314

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

316 The CIM_AccountManagementService provides the ability to manage accounts on the system.
317 CIM_Account represents accounts that are defined locally on the system. CIM_Group provides the ability
318 to group account identities for authorization purposes. CIM_UserContact provides descriptive information
319 about an individual who has been authenticated through third-party authentication. CIM_Identity
320 represents a security principal. The CIM_AssignedIdentity association is used to associate the security
321 principal with the entity whose privileges are being managed. Local accounts, third-party authenticated
322 users, and account identity groups each can have one or more associated security principals. These
323 security principles create a relationship between the authenticated individual and the authorization
324 granted to the individual.

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

327 6.1 Authenticated Entities

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

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

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

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

349 6.2 Account

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

353 6.3 Account States

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

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

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

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

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

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

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

NOTE 3: This state is usually the result of a violation of a system policy. Before access can be granted to the resources secured by the account, corrective action is required in this case.

For example, an account can be placed into the locked-out state because the password expired, the number of consecutive failed access attempts exceeded the limit set by policy, the inactivity period exceeded the limit set by policy, and so on. This action can be taken by the user to whom the account corresponds (for example, changing the password), or it can be an administrative action.

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

EXPERIMENTAL

6.4 Local Account Security Policies

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

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

EXPERIMENTAL

6.5 Access Ingress Point

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

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

6.6 Identity Context

An account, account identity group, or third-party authenticated entity can have more than one security principal associated with it. The security principals are frequently differentiated based on the mechanism

399 through which the credentials that identify the underlying entity were supplied. For example, credentials
400 validated against an account on a system could resolve to a different security principal depending on
401 whether the credentials were supplied over a terminal session, through a remote management interface,
402 or locally. The security principals can have different privileges assigned to them. The need to manage
403 privileges for an authenticated entity that vary based on context is a common reason for having multiple
404 security principals associated with the authenticated entity.

405 7 Implementation

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

408 7.1 Base Requirements

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

410 7.1.1 CIM_AccountManagementService

411 At least one instance of CIM_AccountManagementService shall exist.

412 7.1.1.1 CIM_AccountManagementCapabilities

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

415 7.1.1.1.1 CIM_Account.UserPassword Constraints

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

417 7.1.1.1.1.1 UserPassword Is Clear Text

418 When the SupportedUserPasswordEncryptionAlgorithms[ ] property of
419 CIM_AccountManagementCapabilities is NULL, the CIM_Account.UserPassword property shall be clear
420 text and the CIM_Account.UserPasswordEncryptionAlgorithm property shall have no value.

421 When the SupportedUserPasswordEncryptionAlgorithms[ ] property of
422 CIM_AccountManagementCapabilities has no values , the CIM_Account.UserPassword property shall be
423 clear text and the CIM_Account.UserPasswordEncryptionAlgorithm property shall have no value.

424 When the SupportedUserPasswordEncryptionAlgorithms[ ] property of
425 CIM_AccountManagementCapabilities only has the value 0 (None), the CIM_Account.UserPassword
426 property shall be clear text and the CIM_Account.UserPasswordEncryptionAlgorithm property shall have
427 the value 0 (None).

428 When the SupportedUserPasswordEncryptionAlgorithms[ ] property of
429 CIM_AccountManagementCapabilities has several values, including the value 0 (None), the
430 CIM_Account.UserPassword property may be clear text. In this case when CIM_Account.UserPassword
431 property is in clear text, the CIM_Account.UserPasswordEncryptionAlgorithm property shall have the
432 value 0 (None).

433 7.1.1.1.1.2 UserPassword Is Encrypted

434 When the SupportedUserPasswordEncryptionAlgorithms[ ] property of
435 CIM_AccountManagementCapabilities contains one or more values but not 0 (None), the
436 CIM_Account.UserPassword property shall be encrypted.

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

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

7.1.1.2 CIM_AccountManagementService.ElementName Constraints

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

7.1.1.2.1 ElementName Is Not Modifiable

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

7.1.1.2.2 ElementName Is Modifiable

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

7.1.1.2.2.1 CIM_AccountManagementCapabilities.ElementNameEditSupported

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

7.1.1.2.2.2 CIM_AccountManagementCapabilities.MaxElementNameLen

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

7.1.2 Representing a Security Principal

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

7.1.3 At Least One Authentication Model

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

7.2 Account Creation (Conditional)

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

This section details additional requirements that are conditional on support for account creation. These requirements shall be supported when the CIM_AccountManagementCapabilities.OperationsSupported property of the instance of CIM_AccountManagementCapabilities that is associated with the

CIM_AccountManagementService through the CIM_ElementCapabilities association contains the value 2 (Create).

7.2.1 Modeling Account Defaults (Optional)

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

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

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

7.2.2 Capabilities and Requirements for Account Creation (Optional)

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

7.3 Account Management (Optional)

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

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

7.3.1 Identity for an Account

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

7.3.2 Capabilities of an Account

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

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

7.3.3 Managing the Account's State

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

7.3.3.1 State Management Supported

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

7.3.3.2 CIM_Account.RequestStateChange( ) Supported

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

7.3.3.3 CIM_Account.RequestedState

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

7.3.3.4 CIM_Account.EnabledState

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

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

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

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

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

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

When disabling of an account is supported without the ability to further distinguish between disablement with the clearing of the account configuration and disablement without the clearing of the account

558 configuration, the value 3 (Disabled) shall be used and the value 6 (Enabled but Offline) shall not be
559 used.

560 #### 7.3.3.5 Indicating State Management Support with CIM_EnabledLogicalElementCapabilities

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

565 ### 7.3.4 CIM_Account.ElementName Constraints

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

567 #### 7.3.4.1 ElementName Is Not Modifiable

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

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

572 #### 7.3.4.2 ElementName Is Modifiable

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

576 ##### 7.3.4.2.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

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

579 ##### 7.3.4.2.2 CIM_EnabledLogicalElementCapabilities.MaxElementNameLen

580 The MaxElementNameLen property shall be implemented when the ElementNameEditSupported
581 property has a value of TRUE. The MaxElementNameLen property shall indicate the maximum length of
582 a string that the implementation will accept as a value for the ElementName property of the associated
583 CIM_Account instance.

584 ##### 7.3.4.2.3 CIM_EnabledLogicalElementCapabilities.ElementNameMask

585 The ElementNameMask property shall be implemented when the ElementNameEditSupported property
586 has a value of TRUE. The ElementNameMask property shall contain a regular expression defined using
587 the syntax specified in Annex C of DSP1001.

588

---

589 **EXPERIMENTAL**

590 ### 7.3.5 Modeling Account Requirements and Capabilities (Optional)

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

593 This section describes how constraints for properties of an instance of CIM_Account may be modeled
594 using instances of CIM_AccountSettingData that are associated with an instance of
595 CIM_EnabledLogicalElementCapabilities through an instance of CIM_SettingsDefineCapabilities. One or

596 more instances of CIM_AccountSettingData may be associated with an instance of
597 CIM_EnabledLogicalElementCapabilities through the CIM_SettingsDefineCapabilities association.

598 #### 7.3.5.1 Password History Depth

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

602 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 2 (Maximums), the value
603 of the PasswordHistoryDepth property shall represent the maximum value that is supported for the
604 CIM_Account.PasswordHistoryDepth property.

605 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the value of
606 the PasswordHistoryDepth property shall represent the minimum value that is supported for the
607 CIM_Account.PasswordHistoryDepth property.

608 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 0 (Point), the value of the
609 PasswordHistoryDepth property shall represent the only value that is supported for the
610 CIM_Account.PasswordHistoryDepth property.

611 #### 7.3.5.2 Password Expiration

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

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

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

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

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

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

632 #### 7.3.5.3 Complex Password Rules

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

636 When the CIM_SettingsDefineCapabilities.ValueRange property has the value 3 (Minimums), the values
637 contained in the ComplexPasswordRulesEnforced property shall represent the minimum set of values

that are required to be contained in the CIM_Account.ComplexPasswordRulesEnforced property for the instance of CIM_AccountManagementService that is associated with the CIM_Capabilities instance.

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

7.3.5.4 Inactivity Timeout

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

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

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

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

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

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

NOTE: Account State (see 6.3) may change due to inactivity timeout expiry set by this property.

7.3.5.5 Successive Failed Logins

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

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

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

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

NOTE: Account State (see 6.3) may change after the consecutive failed login attempts set by this property.

EXPERIMENTAL

7.4 Representing a Third-Party Authenticated Principal (Optional)

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

7.4.1 Identity for CIM_UserContact

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

7.4.2 Profile Conformance Scope for CIM_UserContact

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

1) From an instance of CIM_UserContact, traverse the CIM_AssignedIdenity association to locate instances of CIM_Identity.
2) From each found CIM_Identity instance, traverse the CIM_ServiceAffectsElement association to locate instances of CIM_AccountManagementService.

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

7.5 Managing Account Identity Groups (Optional)

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

7.5.1 Managing Local Account Identity Groups

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

7.5.2 Identity for a Group

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

7.5.3 Relating an Account Identity to a Group

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

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

7.6 Representing Access Ingress Point (Optional)

For a particular instance of CIM_Identity, the ingress point through which a currently authenticated session is being maintained may be identified by an **optional** instance of CIM_Dependency. Such an ingress point may be a system, service, protocol endpoint, or other entity through which requests can

719 flow. An instance of CIM_Dependency between an instance of CIM_Identity and an instance of
720 CIM_ManagedElement shall not exist except to represent an authenticated session.

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

722 7.7 Identity Context (Optional)

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

726 The default ingress point for an instance of CIM_Identity is the CIM_System associated with the
727 CIM_AccountManagmentService, (via CIM_HostedService), that manages that instance of CIM_Identity,
728 (as indicated by CIM_ServiceAffectsElement.)

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

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

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

738 NOTE2: The context of an instance of CIM_Identity has no effect on the scope of the privileges (if any) that are
739 granted to the represented security principal. Rather, the context provides information about when one security
740 principal versus another will be selected when credentials are provided that identify an authenticated entity.

741 8 Methods

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

744 8.1 CIM_AccountManagementService.CreateAccount( )

745 The CIM_AccountManagementService.CreateAccount( ) method is used to create accounts on a
746 managed system. When the method returns a value of 0 (zero), a new instance of CIM_Account shall be
747 associated with the CIM_ComputerSystem instance that is identified by the System parameter through
748 the CIM_AccountOnSystem association such that the values of the properties of the instance of
749 CIM_Account are the values of the non-Null properties of the template account instance that is specified
750 by the AccountTemplate parameter. The value of the Account parameter shall be a reference to the
751 instance of CIM_Account. A newly created instance of CIM_Identity shall be associated with the
752 CIM_Account instance through the CIM_AssignedIdentity association. The instance of CIM_Identity shall
753 be associated with the CIM_AccountManagementService through the CIM_ServiceAffectsElement
754 association.

755 When the CIM_ComputerSystem instance identified by the System parameter is not associated with the
756 CIM_AccountManagementService instance through the CIM_HostedService association, the method
757 shall return the value 2.

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

760 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

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

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

763 ### 8.1.1 Account Template Requirements

764 This section details the requirements for the AccountTemplate parameter.

765 When the AccountTemplate embedded instance contains the UserPasswordEncryptionAlgorithm property
766 and the value specified for the property is not a supported value as defined in section 7.1.1.1.1 the
767 method shall return the value 2.

768 When the AccountTemplate embedded instance contains the UserPassword property and the value
769 specified for the property is not a supported value as defined in section 7.1.1.1.1 the method shall return
770 the value 2.

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

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

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

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

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

786 If the AccountTemplate embedded instance contains the LastLogin property, the value specified shall be
787 ignored.

8.1.2 Account Default Values

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

8.1.2.1 Using a Default Configuration

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

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

8.1.2.2 Using Implicit Defaults

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

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

8.1.3 CIM_AccountManagementService.CreateAccount( ) Conditional Support

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

8.2 CIM_Account.RequestStateChange( )

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

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

A value of 6 (Offline) shall place the account into the offline state.

When the RequestedState parameter has the value 2 (Enabled), the method may return the value 2 if the account is not properly configured.

See section 7.3.3.3 for information about the effect of this method on the RequestedState property.

The method shall be considered successful if the availability of the functionality upon completion of the method corresponds to the desired availability indicated by the RequestedState parameter. It is not necessary that an actual change in state occur for the method to be considered successful. It is sufficient

829 that the resultant state be equal to the requested state. Upon successful completion of the method, the
830 Return Value shall be 0 (zero).

831 See section 7.3.3.4 for information about the effect of this method on the EnabledState property.

832 RequestStateChange( ) method return code values shall be as specified in Table 4.
833 RequestStateChange( ) method parameters are specified in Table 5.

834 No standard messages are defined.

835 Invoking the CIM_Account.RequestStateChange( ) method multiple times could result in earlier requests
836 being overwritten or lost.

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

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

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

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

839 ### 8.2.1 CIM_Account.RequestStateChange( ) Conditional Support

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

843 ## 8.3 Profile Conventions for Operations

844 Support for operations for each profile class (including associations) is specified in the following
845 subclauses. Each subclause includes either the statement “All operations in the default list in section 8.3
846 are supported as described by DSP0200 version 1.2” or a table listing all of the operations that are not
847 supported by this profile or where the profile requires behavior other than that described by DSP0200
848 version 1.2.

849 The default list of operations is as follows:

850 • GetInstance
851 • Associators

852 • AssociatorNames
853 • References
854 • ReferenceNames
855 • EnumerateInstances
856 • EnumerateInstanceNames

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

859 ## 8.4 CIM_Account

860 Table 6 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or
861 shall not be supported.

862 **Table 6 – Operations: CIM_Account**

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

863 ### 8.4.1 CIM_Account—GetInstance Operation

864 The following are possible behaviors and mutually exclusive:

865 • When the GetInstance operation is executed against an instance of CIM_Account and the
866 underlying account has a valid password, the value of the CIM_Account.UserPassword property
867 shall be an array of length zero to indicate that the account has a password configured and is
868 unable or unwilling to return the value in clear text.

869 • When the GetInstance operation is executed against an instance of CIM_Account and the
870 underlying account does not have a valid password, the CIM_Account.UserPassword property
871 shall be Null.

872 ### 8.4.2 CIM_Account—ModifyInstance Operation

873 The ModifyInstance operation shall be supported if and only if the
874 OperationsSupported property contains the value 3 (Modify) for an instance of
875 CIM_AccountManagementCapabilities that is associated through the
876 CIM_ElementCapabilities association with an instance of
877 CIM_AccountManagementService associated through CIM_ServiceAffectsElement with an instance of
878 CIM_Identity that is associated with the instance of CIM_Account through CIM_AssignedIdentity.

879 As described in 7.1.1.1.1 the UserPassword property of CIM_Account may be in clear text or be
880 encrypted. Encrypting UserPassword may be required since the network session may not be encrypted.

881 When the ModifyInstance operation is supported and a value is specified for the
882 CIM_Account.UserPassword property and the CIM_Account.UserPasswordEncryptionAlgorithm property
883 has no value or has the value 0 (None), the value of the CIM_Account.UserPassword property shall be
884 clear text without encryption.

885 When the ModifyInstance operation is supported and a value is specified for the
886 CIM_Account.UserPassword property and the CIM_Account.UserPasswordEncryptionAlgorithm property

887 has a non-zero value, the value of the CIM_Account.UserPassword property shall be encrypted in the
888 form specified by the value of the CIM_Account.UserPasswordEncryptionAlgorithm property

889 8.4.3 CIM_Account—DeleteInstance Operation

890 The DeleteInstance operation shall be supported if and only if the OperationsSupported property contains
891 the value 4 (Delete) for an instance of CIM_AccountManagementCapabilities that is associated through
892 the CIM_ElementCapabilities association with an instance of CIM_AccountManagementService
893 associated through CIM_ServiceAffectsElement with an instance of CIM_Identity that is associated with
894 the instance of CIM_Account through CIM_AssignedIdentity.

895 When the associated instance of CIM_Identity is not associated with any other instances of
896 CIM_ManagedElement through the CIM_AssignedIdentity association, the CIM_Identity instance shall be
897 deleted.

898 When the associated instance of CIM_EnabledLogicalElementCapabilities is not associated with any
899 other instance of CIM_Account through the CIM_ElementCapabilities association, the instance of
900 CIM_EnabledLogicalElementCapabilities shall be deleted.

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

902 8.5 CIM_EnabledLogicalElementCapabilities

903 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

904 8.6 CIM_AccountOnSystem

905 Table 7 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or
906 shall not be supported.

907 **Table 7 – Operations: CIM_AccountOnSystem**

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

908 8.7 CIM_AccountManagementCapabilities

909 All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

910 8.8 CIM_AccountManagementService

911 Table 8 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or
912 shall not be supported.

913 **Table 8 – Operations: CIM_AccountManagementService**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.8.1.	None

8.8.1 CIM_AccountManagementService—ModifyInstance Operation

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

8.8.1.1 CIM_AccountManagementService.ElementName property

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

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

8.9 CIM_AccountSettingData

Table 9 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 9 – Operations: CIM_AccountSettingData

Operation	Requirement	Messages
ModifyInstance	Optional	None

8.10 CIM_AssignedIdentity

Table 10 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 10 – Operations: CIM_AssignedIdentity

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

8.11 CIM_Dependency

Table 11 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 11 – Operations: CIM_Dependency

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

8.12 CIM_ElementCapabilities

Table 12 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 12 – Operations: CIM_ElementCapabilities

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

8.13 CIM_ElementSettingData

Table 13 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 13 – Operations: CIM_ElementSettingData

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

8.13.1 CIM_ElementSettingData—ModifyInstance

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

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

1) The ModifyInstance operation may find another instance of CIM_ElementSettingData that associates an instance of CIM_AccountSettingData with the instance of CIM_AccountManagementService that is referenced by the target instance of CIM_ElementSettingData where the IsNext property has a value of 1 (Is Next).

2) For the instance of CIM_ElementSettingData found, the ModifyInstance operation shall modify the value of its IsNext property to have a value of 2 (Is Not Next).

8.14 CIM_Group

All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

8.15 CIM_HostedService

Table 14 lists operations that either have special requirements beyond those from DSP0200 version 1.2 or shall not be supported.

Table 14 – Operations: CIM_HostedService

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

8.16 CIM_Identity

All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

8.17 CIM_IdentityContext

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_IdentityContext

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

8.18 CIM_MemberOfCollection

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_MemberOfCollection

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

8.18.1 CIM_MemberOfCollection—CreateInstance

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

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

8.18.2 CIM_MemberOfCollection—DeleteInstance

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

8.19 CIM_OwningCollectionElement

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_OwningCollectionElement

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

8.20 CIM_ServiceAffectsElement

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_ServiceAffectsElement

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

8.21 CIM_SettingsDefineCapabilities

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_SettingsDefineCapabilities

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

8.22 CIM_UserContact

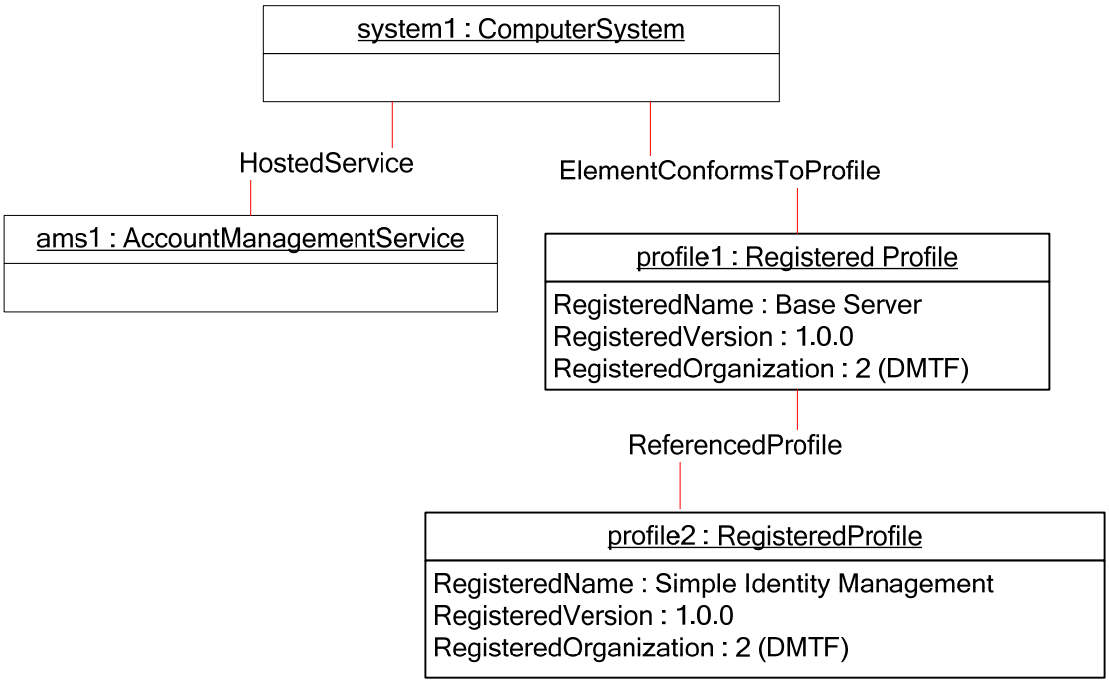
All operations in the default list in section 8.3 are supported as described by DSP0200 version 1.2.

9 Use Cases

This section contains object diagrams and use cases for the *Simple Identity Management 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 *Simple Identity Management Profile*. Using scoping instance methodology as described in the *Profile Registration Profile*, profile2 contains the version information for the *Simple Identity Management Profile* implementation.

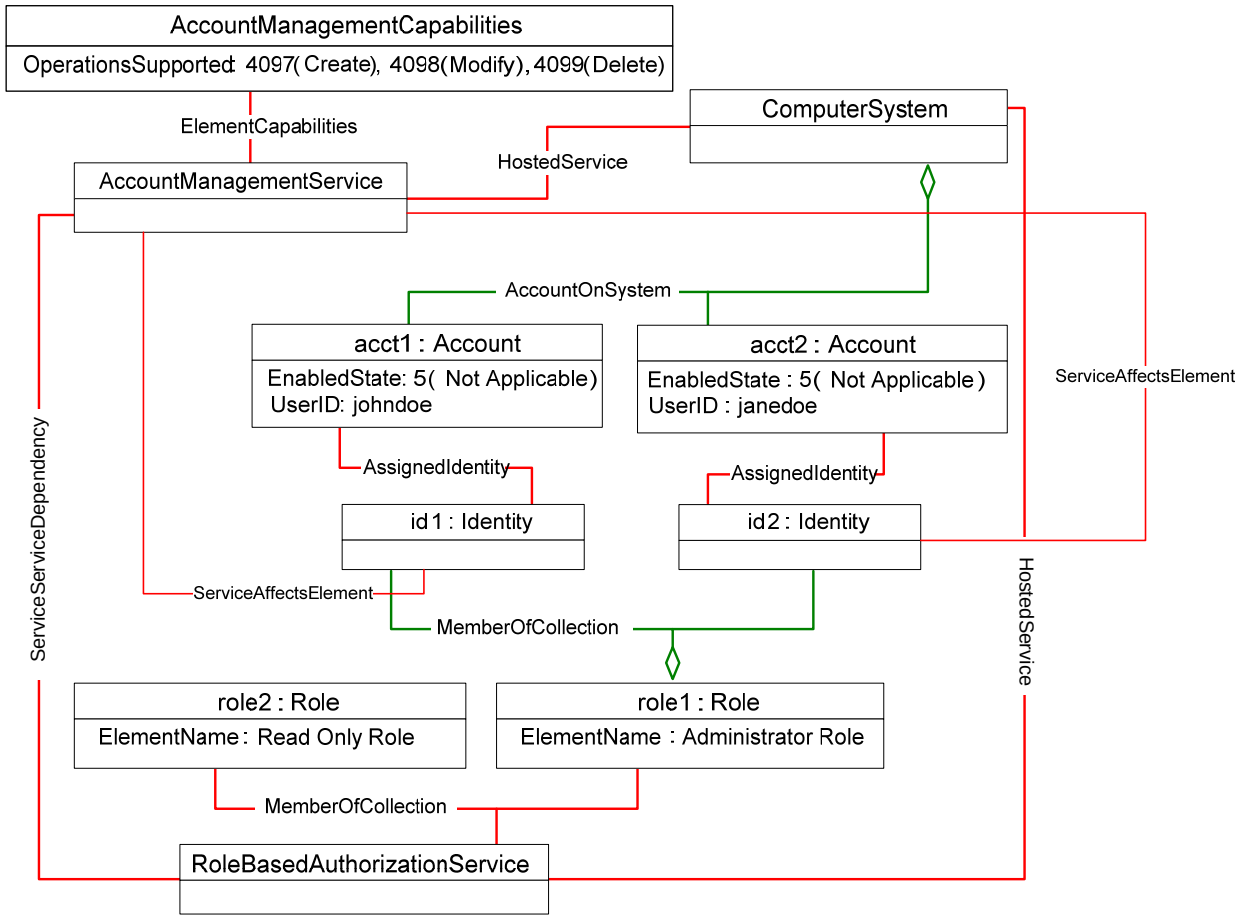


1013

1014 **Figure 2 – Profile Registration**

1015 Figure 3 shows a system that supports management of local accounts for authentication and
1016 authorization. The modeled system supports creation, modification, and deletion of accounts. Privilege
1017 management is performed through assignment to Roles.

1018

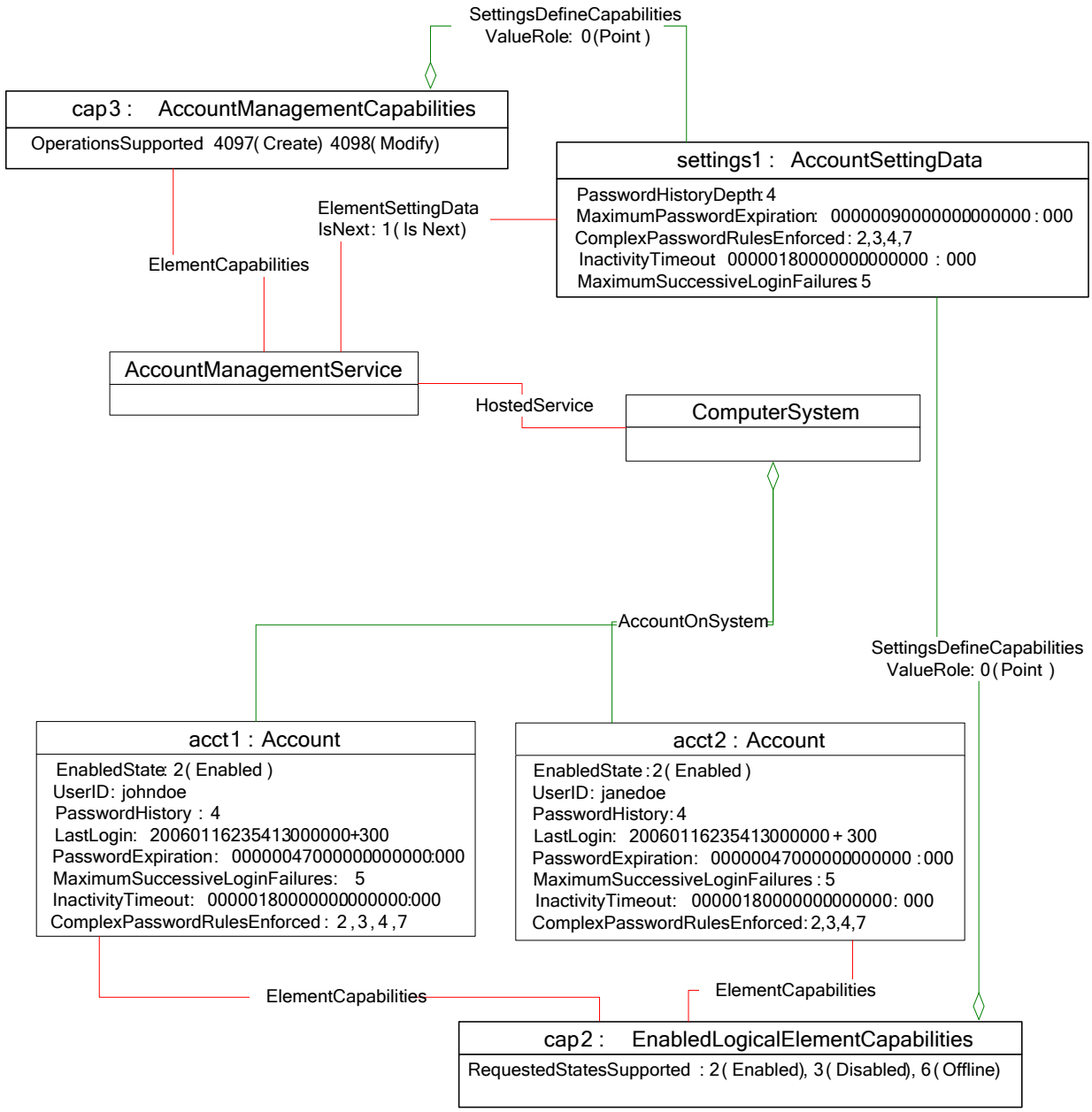


1019

1020 **Figure 3 – Basic System Accounts**

1021 Figure 4 shows a system that supports capabilities related to password management. Accounts created
1022 through the CIM_AccountManagementService are required to maintain a history of the four previous
1023 passwords, have the password changed every 90 days, enter a locked-out state after 180 days of
1024 inactivity, and enter a locked-out state after five successive failed login attempts. Additionally, passwords
1025 are required to have a minimum length, not contain the user ID, contain at least one numeric character,
1026 and enforce a maximum number of repeating characters. These requirements are indicated by the
1027 CIM_SettingsDefineCapabilities association between settings1 and cap3.

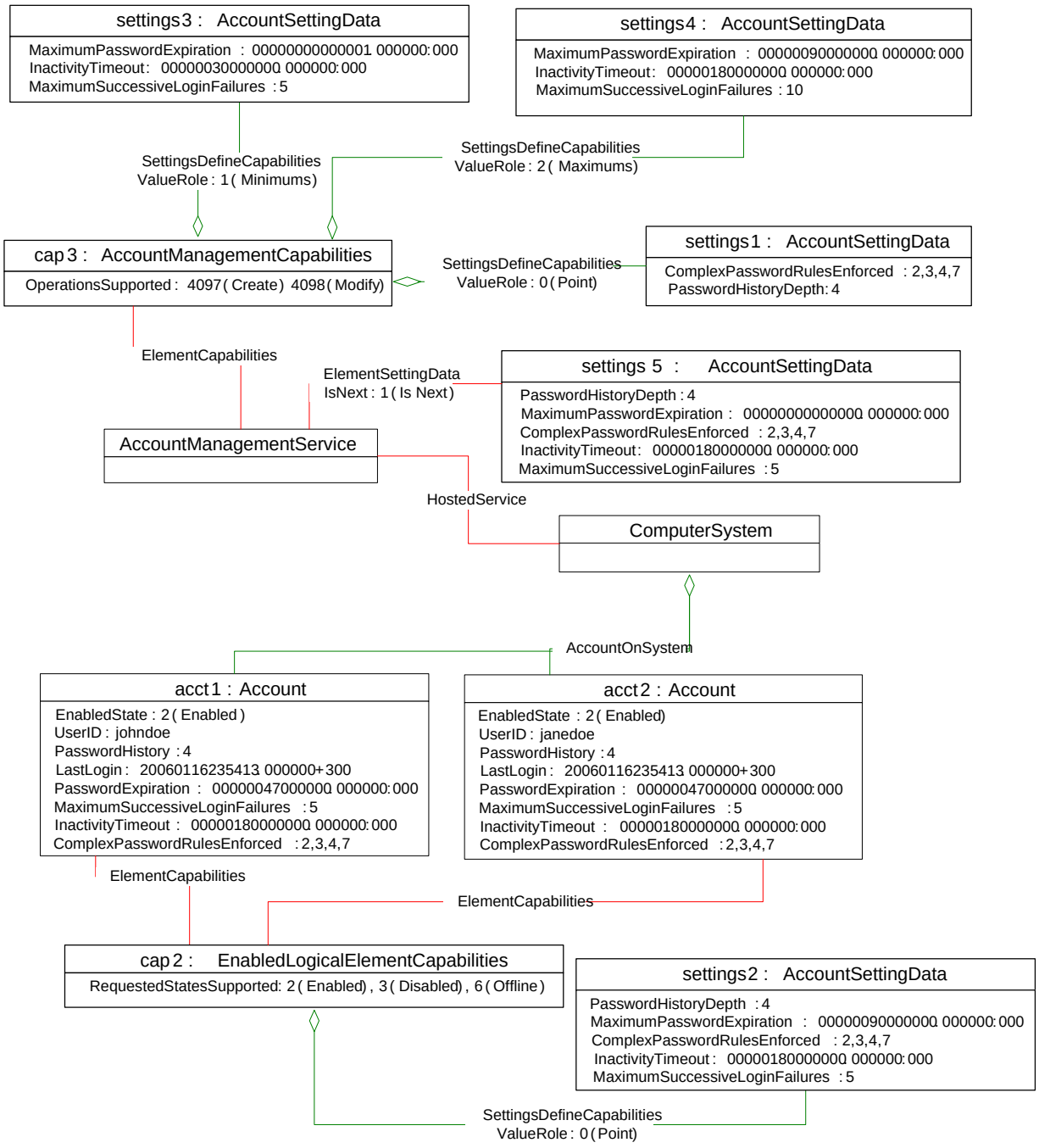
1028 acct1 and acct2 operate under the same password constraints as new accounts created through the
1029 CIM_AccountManagementService. This behavior is indicated by the CIM_SettingsDefineCapabilities
1030 association between cap2 and settings1. The password for each account is required to be changed every
1031 90 days. Each account currently has 47 days until the password needs to be changed. Thus the
1032 password for each account was last changed 43 days ago. Similarly, the accounts are required to enter a
1033 locked-out state after 180 days of inactivity. Each account currently has 180 days until it will be locked.
1034 Therefore each account was last accessed today.



1035

1036 **Figure 4 – Full Account Capabilities**

Figure 5 also shows a system that supports capabilities related to password management. Accounts created through the CIM_AccountManagementService are required to maintain a history of the four previous passwords. Account passwords are required to be changed at least every 90 days. The inactivity timeout can be configured to be from 30 to 180 days. The number of successive failed login attempts can be configured to be between five and ten. Additionally, passwords are required to have a minimum length, not contain the user ID, contain at least one numeric character, and enforce a maximum number of repeating characters. These constraints are indicated by the CIM_SettingsDefineCapabilities association between cap3 and settings1, settings3, and settings4. acct1 and acct2 operate under the same password constraints. These constraints are within the range allowed for created accounts. These constraints are indicated by the CIM_SettingsDefineCapabilities association between cap2 and settings2. The password for each account is required to be changed every 90 days. Each account currently has 47 days until the password needs to be changed. Thus, the password for each account was last changed 43 days ago. Similarly, the accounts are required to enter a locked-out state after 180 days of inactivity. Each account currently has 180 days until it will be locked. Therefore each account was last accessed today. AccountSettingData settings5 shows the default setting.



1052

1053 **Figure 5 – Account Capabilities with Ranges**

1054

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

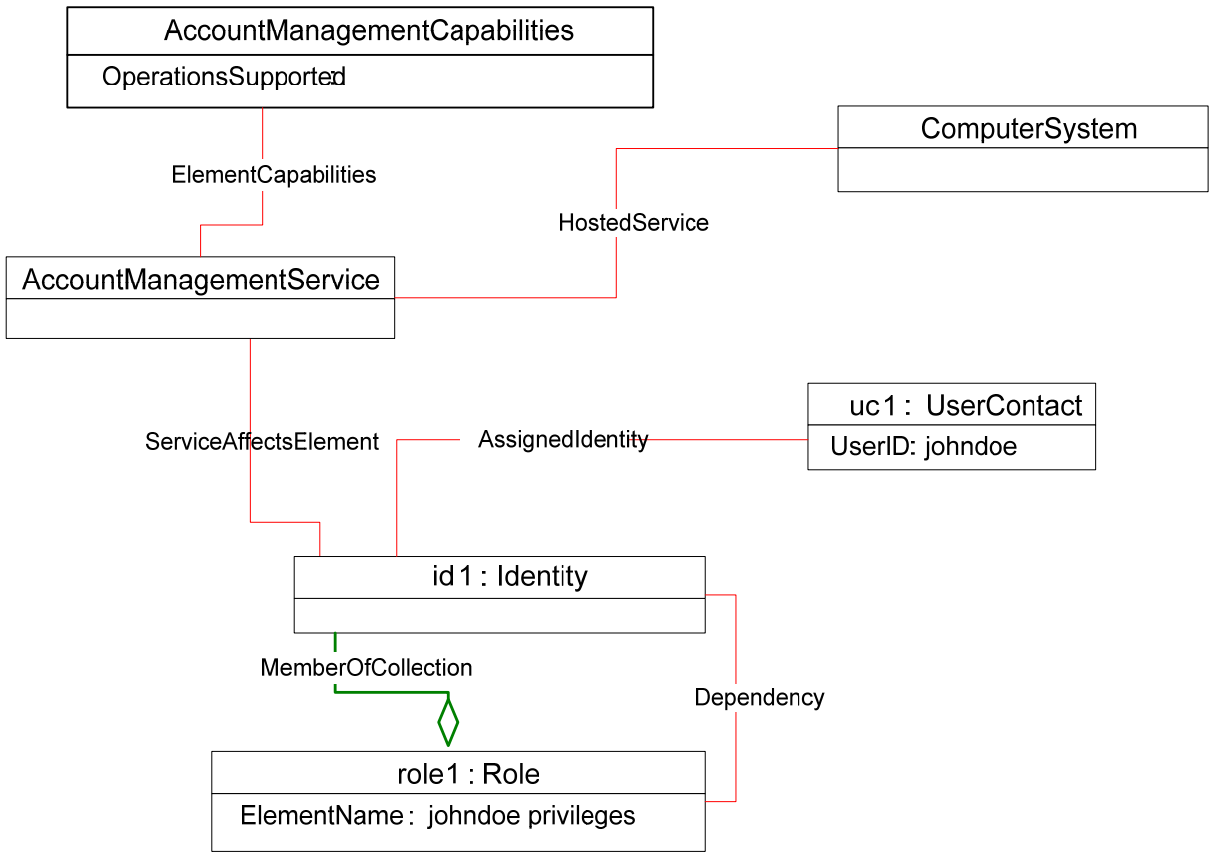
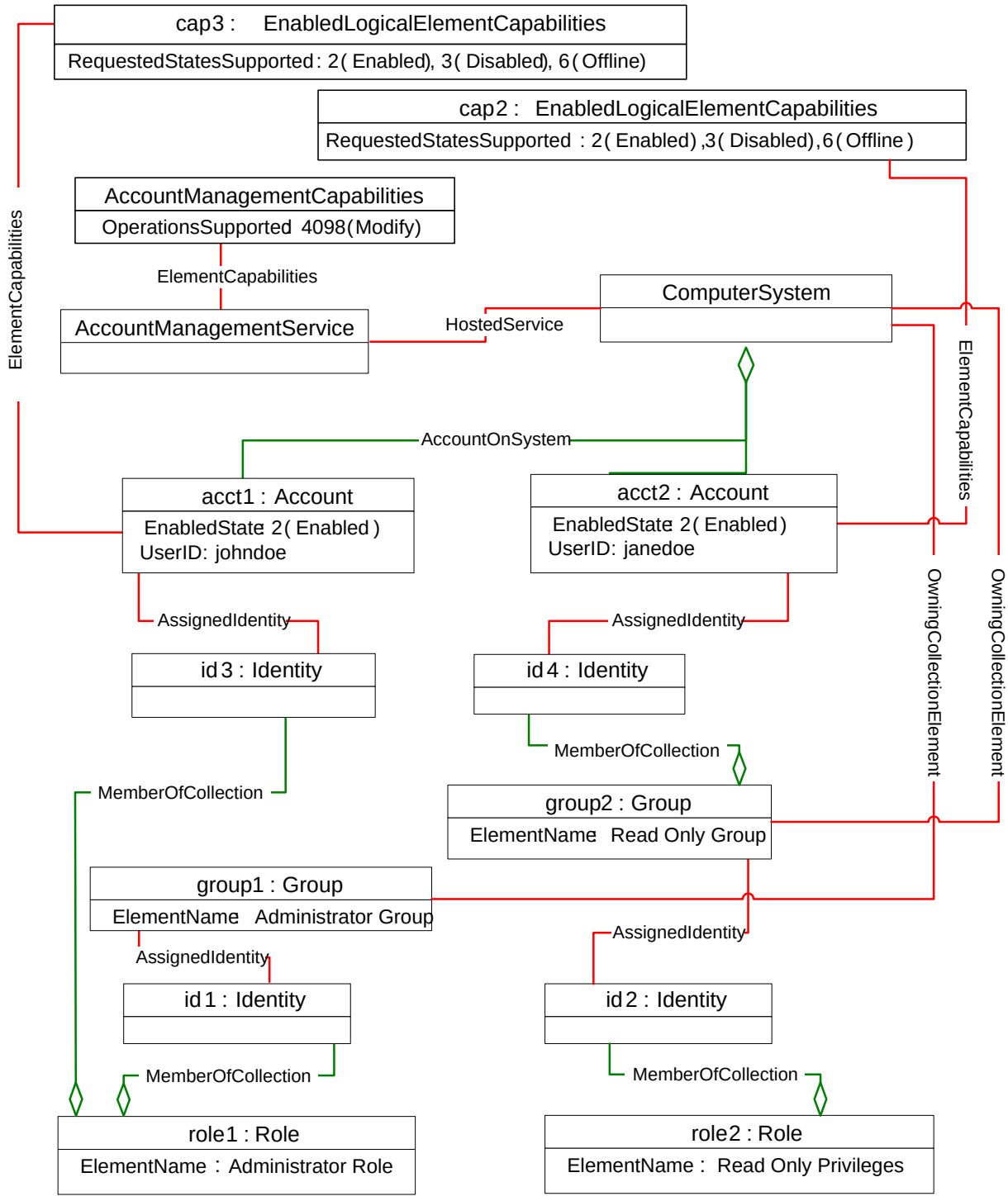


Figure 6 – Third-Party Authenticated User

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

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

1074



1075

1076 Figure 7 – Accounts with Group Membership

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

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

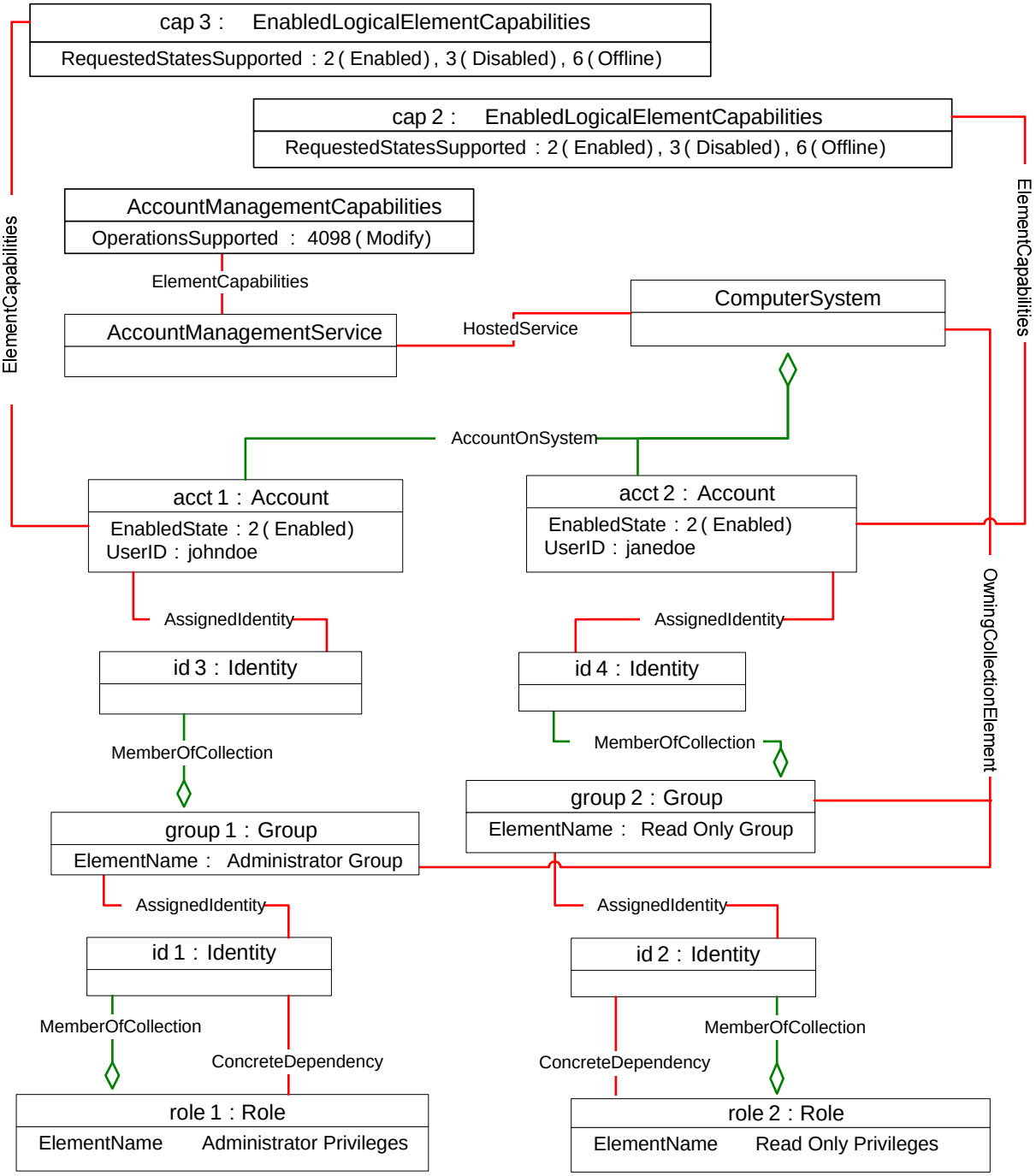


Figure 8 – Role-Oriented Groups

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

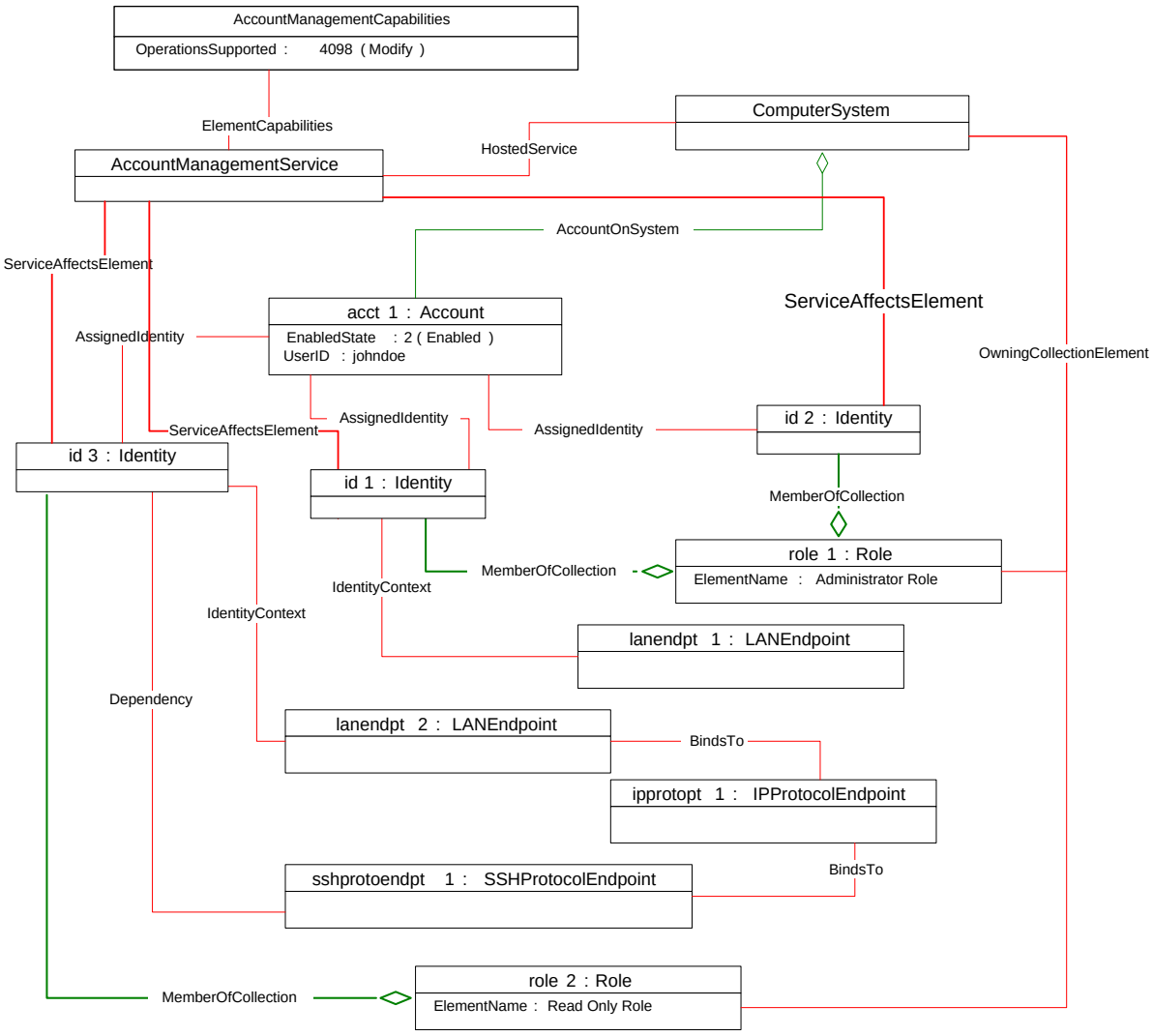


Figure 9 – Access Ingress Point and Identity Context

9.2 Determine Whether CIM_Account.ElementName Can Be Modified

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

1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the target instance.

3) Query the value of the ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities instance.

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

9.3 Determine Whether Account State Management Is Supported

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

1) Find the CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_Account instance.

4) Query the value of the RequestedStatesSupported property.

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

9.4 Determine Whether Account Management Is Supported

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

1) Starting at the CIM_ComputerSystem instance for the managed system, look for an instance of CIM_AccountManagementService with which it is associated through the CIM_HostedService association.

5) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.

6) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

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

9.5 Create an Account

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

1) Determine if account creation is supported as follows:

    1) Starting at the CIM_ComputerSystem instance for the managed system, look for an instance of CIM_AccountManagementService with which it is associated through the CIM_HostedService association.

    2) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.

    3) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

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

7) Create a template instance of CIM_Account.

8) Invoke the CIM_AccountManagementService.CreateAccount( ) method, specifying the template instance.

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

9.6 Determine Account Defaults

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

1) Starting with the CIM_AccountManagementService, look for an instance of CIM_AccountSettingData with which it is associated through the CIM_ElementSettingData association where the CIM_ElementSettingData.IsNext property has the value 1 (Is Next).

9) If an instance is found, query the values of the properties to determine the default configuration.

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

9.7 Delete an Account

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

1) Determine if account deletion is supported as follows:

   1) Starting at the CIM_Account instance, look for an instance of CIM_AccountManagementService with which it is associated. CIM_Account is associated with CIM_Identity through the CIM_AssignedIdentity association and CIM_Identity is associated with the AccountManagementService through the CIM_ServiceAffectsElement association

   2) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.

   3) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

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

10) Invoke the DeleteInstance operation against the instance of CIM_Account.

9.8 Modify the Password for an Account

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

1) Determine if account modification is supported as follows:

   1) Starting at the CIM_Account instance, look for an instance of CIM_AccountManagementService with which it is associated. CIM_Account is associated with CIM_Identity through the CIM_AssignedIdentity association and CIM_Identity is associated with the AccountManagementService through the CIM_ServiceAffectsElement association

   2) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.

   3) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

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

11) Invoke the GetInstance operation against the target instance of CIM_Account

12) Modify the UserPassword property.

13) Invoke the ModifyInstance operation.

9.9 Clear an Account

A client can clear an account as follows:

1) Starting at the instance of CIM_Account, look for an instance of CIM_EnabledLogicalElementCapabilities with which it is associated through the CIM_ElementCapabilities association.

14) If an instance is found, query the RequestedStatesSupported property to determine if it contains the value 3 (Disabled).

15) Invoke the CIM_Account.RequestStateChange( ) method specifying a value of 3 (Disabled).

9.10 Change state to Enabled Offline

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

1) Starting at the instance of CIM_Account, look for an instance of CIM_EnabledLogicalElementCapabilities with which it is associated through the CIM_ElementCapabilities association.

16) If an instance is found, query the RequestedStatesSupported property to determine if it contains the value 6 (Enabled but Offline).

17) Invoke the CIM_Account.RequestStateChange( ) method specifying a value of 6 (Enabled but Offline).

9.11 Add an Account Identity to a Group

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

1) Find an instance of CIM_Identity that is associated with the target instance of CIM_Account through the CIM_AssignedIdentity association.

2) Invoke the CreateInstance operation against CIM_MemberOfCollection where the template instance references the desired instances of CIM_Identity and CIM_Group.

9.12 Remove an Account Identity from a Group

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

1) Find each instance of CIM_Identity that is associated with the target CIM_Account instance through the CIM_AssignedIdentity association.

2) For each instance of CIM_Identity, test whether it is associated with the target CIM_Group instance through the CIM_MemberOfCollection association.

3) If the instance of CIM_MemberOfCollection exists, execute the DeleteInstance operation against it.

9.13 Determine the Context of a Security Principal

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

10 CIM Elements

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

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

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

10.1 CIM_Account

Table 21 details the requirements for instances of CIM_Account.

Table 21 – Class: CIM_Account

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
UserID	Mandatory	(pattern ".*")
UserPassword	Mandatory	(pattern ".*")
OrganizationName	Mandatory	(pattern ".*")
ElementName	Mandatory	See section 7.3.4.1
UserPasswordEncryptionAlgorithm	Optional	See section 7.1.1.1.1
OtherUserPasswordEncryptionAlgorithm	Conditional	Mandatory when UserPasswordEncryptionAlgorithm is 1(Other).
PasswordHistoryDepth	Optional	EXPERIMENTAL, See section 7.3.5.1.
PasswordExpiration	Optional	EXPERIMENTAL, See section 7.3.5.2.
ComplexPasswordRulesEnforced	Optional	EXPERIMENTAL, See section 7.3.5.3.
InactivityTimeout	Optional	EXPERIMENTAL, See section 7.3.5.4.
MaximumSuccessiveLoginFailures	Optional	EXPERIMENTAL, See section 7.3.5.5.
RequestedState	Mandatory	See section 7.3.3.3.
EnabledState	Mandatory	See section 7.3.3.4.
RequestStateChange()	Conditional	See section 7.3.3.2.

10.2 CIM_AccountManagementCapabilities

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

Table 22 – Class: CIM_AccountManagementCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	None
ElementNameEditSupported	Mandatory	See section 7.3.4.2.1.
MaxElementNameLen	Conditional	See section 7.3.4.2.2.
ElementName	Mandatory	pattern ".*"
OperationsSupported	Mandatory	None
SupportedUserPasswordEncryptionAlgorithms[ ]	Optional	See section 7.1.1.1.

10.3 CIM_AccountManagementService

Table 23 details the requirements for instances of CIM_AccountManagementService.

Table 23 – Class: CIM_AccountManagementService

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

10.4 CIM_AccountOnSystem

Table 24 details the requirements for instances of CIM_AccountOnSystem. The existence of CIM_AccountOnSystem is conditional on the existence of an instance of CIM_Account.

Table 24 – Class: CIM_AccountOnSystem

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

10.5 CIM_AccountSettingData

Table 25 details the requirements for instances of CIM_AccountSettingData.

Table 25 – Class: CIM_AccountSettingData

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

10.6 CIM_AssignedIdentity (Account)

Table 26 details the requirements for instances of CIM_AssignedIdentity. The existence of CIM_AssignedIdentity used in this context is conditional on an instance of CIM_Identity for a CIM_Account.

Table 26 – Class: CIM_AssignedIdentity (Account)

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

10.7 CIM_AssignedIdentity (Group)

Table 27 details the requirements for instances of CIM_AssignedIdentity. The existence of CIM_AssignedIdentity used in this context is conditional on an instance of CIM_Identity for a CIM_Group.

Table 27 – Class: CIM_AssignedIdentity (Group)

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

10.8 CIM_AssignedIdentity (UserContact)

Table 28 details the requirements for instances of CIM_AssignedIdentity. The existence of CIM_AssignedIdentity used in this context is conditional on an instance of CIM_Identity for a CIM_UserContact.

Table 28 – Class: CIM_AssignedIdentity (UserContact)

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

10.9 CIM_Dependency (Access Ingress)

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

Table 29 – Class: CIM_Dependency (Access Ingress)

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

10.10 CIM_ElementCapabilities (CIM_AccountManagementService)

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

Table 30 – Class: CIM_ElementCapabilities (CIM_AccountManagementService)

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

10.11 CIM_ElementCapabilities (CIM_Account)

CIM_ElementCapabilities associates an instance of CIM_EnabledLogicalElementCapabilities with an instance of CIM_Account. Table 31 details the requirements for instances of CIM_ElementCapabilities. This instance of CIM_Account is conditional on the instrumentation of an instance of CIM_EnabledLogicalElementCapabilities for the CIM_Account instance.

Table 31 – Class: CIM_ElementCapabilities (CIM_Account)

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

10.12 CIM_ElementSettingData

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

Table 32 – Class: CIM_ElementSettingData

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

10.13 CIM_EnabledLogicalElementCapabilities

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

Table 33 – Class: CIM_EnabledLogicalElementCapabilities

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

10.14 CIM_Group

Table 34 details the requirements for instances of CIM_Group.

Table 34 – Class: CIM_Group

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

10.15 CIM_HostedService

Table 35 details the requirements for instances of CIM_HostedService.

Table 35 – Class: CIM_HostedService

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

10.16 CIM_Identity

Table 36 details the requirements for instances of CIM_Identity. Note that CIM_Identity is optional even though at least one instance is required to be associated with an instance of CIM_Account because CIM_Account is itself optional.

Table 36 – Class: CIM_Identity

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

10.17 CIM_IdentityContext

Table 37 details the requirements for instances of CIM_IdentityContext.

Table 37 – Class: CIM_IdentityContext

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

10.18 CIM_MemberOfCollection (Group Membership)

Table 38 details the requirements for instances of CIM_MemberOfCollection when it is used to associate instances of CIM_Identity with instances of CIM_Group. The existence of CIM_MemberOfCollection is conditional on the existence of an instance of CIM_Group and at least one instance of CIM_Identity assigned to a CIM_Group.

1305 **Table 38 – Class: CIM_MemberOfCollection (Group Membership)**

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

1306 ## 10.19 CIM_OwningCollectionElement

1307 Table 39 details the requirements for instances of CIM_OwningCollectionElement. The existence of an
1308 instance of CIM_OwningCollectionElement is conditional on the existence of an instance of CIM_Group.

1309 **Table 39 – Class: CIM_OwningCollectionElement**

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

1310 ## 10.20 CIM_ServiceAffectsElement

1311 Table 40 details the requirements for instances of CIM_ServiceAffectsElement.

1312 **Table 40 – Class: CIM_ServiceAffectsElement (Account)**

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

1313 ## 10.21 CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)

1314 Table 41 details the requirements for instances of CIM_SettingsDefineCapabilities when it is used to
1315 associate an instance of CIM_AccountSettingData with an instance of
1316 CIM_AccountManagementCapabilities. The value of the PropertyPolicy property is fixed at 0
1317 (Independent), which indicates that the value of each property on the referenced
1318 CIM_AccountSettingData instances is independent of the values of the other properties. The ValueRole[]
1319 property is fixed at the value 3 (Supported), which indicates that the value of each property on a
1320 referenced instance of CIM_AccountSettingData represents an inclusive constraint.

1321 **Table 41 – Class: CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)**

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

1322 ## 10.22 CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)

1323 Table 42 details the requirements for instances of CIM_SettingsDefineCapabilities when it is used to
1324 associate an instance of CIM_AccountSettingData with an instance of
1325 CIM_EnabledLogicalElementCapabilities. The value of the PropertyPolicy property is fixed at 0
1326 (Independent), which indicates that the value of each property on the referenced
1327 CIM_AccountSettingData instances is independent of the values of the other properties. The ValueRole[]
1328 property is fixed at the value 3 (Supported), which indicates that the value of each property on a
1329 referenced instance of CIM_AccountSettingData represents an inclusive constraint.

1330 **Table 42 – Class: CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)**

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

1331 ## 10.23 CIM_UserContact

1332 Table 43 details the requirements for instances of CIM_UserContact.

1333 **Table 43 – Class: CIM_UserContact**

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

10.24 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Simple Identity Management Profile*. The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in Table 44, the behavior of the CIM_RegisteredProfile instance is defined by the *Profile Registration Profile*.

Table 44 – Class: CIM_RegisteredProfile

Elements	Requirement	Notes
RegisteredName	Mandatory	Matches "Simple Identity Management"
RegisteredVersion	Mandatory	Matches "1.0.0"
RegisteredOrganization	Mandatory	Matches 2 ("DMTF")

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.0.0a	10/30.2006	Aaron Merkin	Preliminary Standard
1.0.0b	07/31/2007	Murali Rajagopal	Incorporate all Accepted Ballot comments
1.0.0c	11/16/2007	Murali Rajagopal	Incorporate all Accepted Ballot comments
1.0.0d	02/27/2008	Murali Rajagopal	Added EXPERIMENTAL tags
1.0.0.000	03/27/2008	Murali Rajagopal	Ballot to go Final
1.0.0.001	04/08/2008	Murali Rajagopal	Incorporated ballot comments
1.0.0.002	05/01/2008	Murali Rajagopal	Incorporated ballot comments

ANNEX B
(informative)

Acknowledgements

The authors wish to acknowledge the following people.

Authors:

- Aaron Merkin – IBM
- Murali Rajagopal – Broadcom

Contributors from the DMTF WBEM Infrastructure and Protocols Working Group:

- Hemal Shah – Broadcom
- Jon Hass – Dell
- Khachatur Papanyan – Dell
- George Ericson – EMC
- Christina Shaw – HP
- David Hines – Intel