Document Number: DSP1034

Date: 2012-12-13

Version: 1.1.0

Simple Identity Management Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: en-US

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

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

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

CONTENTS

Figures

Tables

Foreword

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

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

Acknowledgments

The authors wish to acknowledge the following people.

Authors:

- Aaron Merkin – IBM
- Murali Rajagopal – Broadcom
- Hemal Shah – Broadcom
- Jim Davis – WBEM Solutions

Contributors:

- Jon Hass – Dell
- Khachatur Papanyan – Dell
- George Ericson – EMC
- Christina Shaw – Hewlett-Packard Company
- David Hines – Intel

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.

221 # Simple Identity Management Profile

222 ## 1 Scope

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

227 ## 2 Normative References

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

231 ### 2.1 Approved References

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

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

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

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

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

242 ANSI T1.276-2003, *Operations, Administration, Maintenance, and Provisioning Security Requirements for*
243 *the Public Telecommunications Network: A Baseline of Security Requirements for the Management*
244 *Plane,* http://webstore.ansi.org

245 ### 2.2 Other References

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

248 ## 3 Terms and Definitions

249 For the purposes of this document, the following terms and definitions apply. For the purposes of this
250 document, the terms and definitions given in DSP1033, DSP1001, and T1.276-2003 also apply.

251 **3.1**
252 **account identity**
253 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

296 **3.14**
297 **unspecified**
298 indicates that this profile does not define any constraints for the referenced CIM element or operation

299 **3.15**
300 **authentication**
301 the process of verifying the credentials provided by an entity for the purpose of resolving to a security
302 principal

303 **3.16**
304 **first-party authentication**
305 authentication that is performed using services that execute local to the relying party

306 **3.17**
307 **principal**
308 an entity that can be positively identified and verified through an authentication mechanism

309 **3.18**
310 **third-party authentication**
311 authentication that is performed using services that execute remote to the relying party

312 4 Symbols and Abbreviated Terms

313 The following abbreviations are used in this document.

314 **4.1**
315 **CIM**
316 Common Information Model

317 5 Synopsis

318 **Profile Name:** *Simple Identity Management*

319 **Version:** 1.1.0

320 **Organization:** DMTF

321 **CIM schema version:** 2.35

322 **Central Class:** CIM_AccountManagementService

323 **Scoping Class:** CIM_ComputerSystem

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

326 CIM_AccountManagementService shall be the Central Class of this profile. The instance of
327 CIM_AccountManagementService shall be the Central Instance of this profile. CIM_ComputerSystem
328 shall be the Scoping Class of this profile. The instance of CIM_ComputerSystem with which the Central
329 Instance is associated through an instance of CIM_HostedService shall be the Scoping Instance of this
330 profile.

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

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

6 Description

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

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

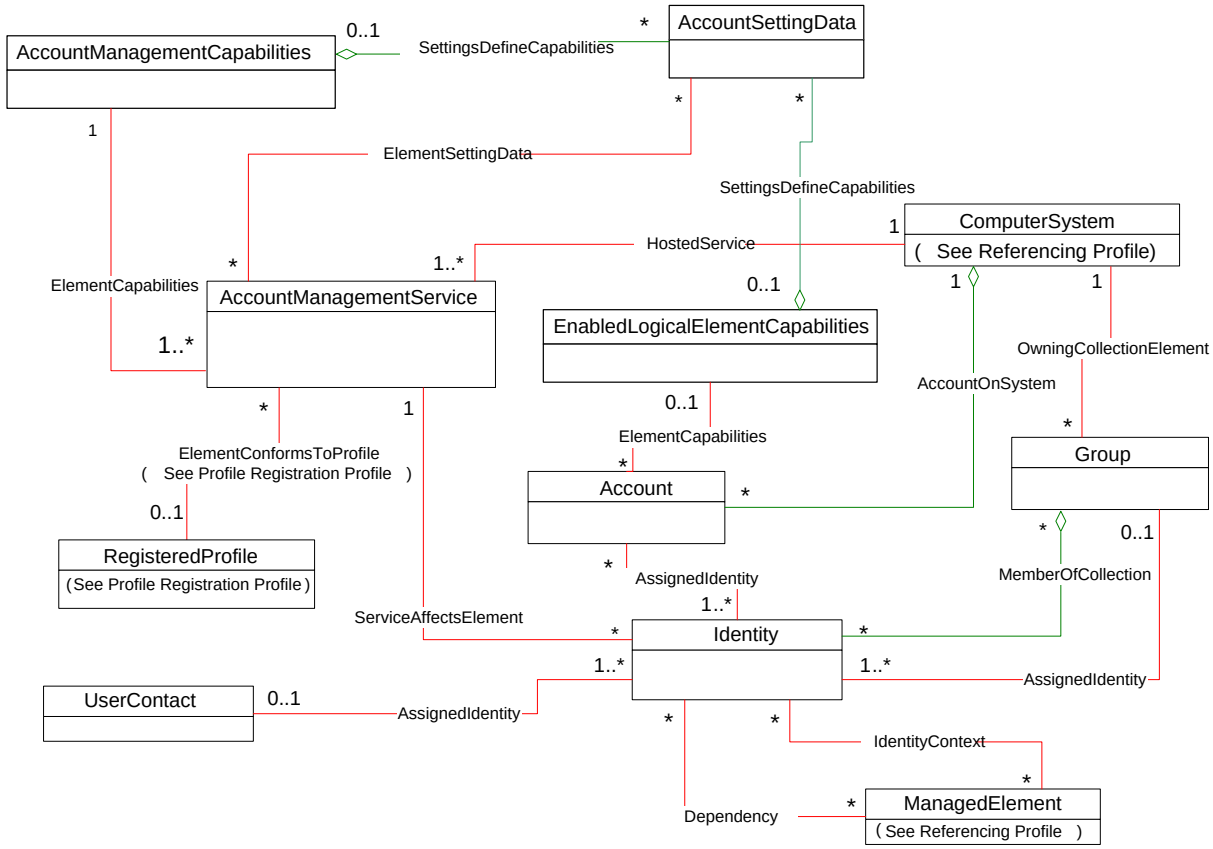


Figure 1 – *Simple Identity Management Profile*: Class Diagram

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

348 security principles create a relationship between the authenticated individual and the authorization
349 granted to the individual.

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

352 6.1 Authenticated Entities

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

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

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

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

374 6.2 Account

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

378 6.3 Account States

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

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

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

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

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

389 NOTE 2: Some accounts may enter the offline state from the disabled state before entering the enabled state. Some
390 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, a changing the password), or it can be an administrative action.

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

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.

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

7 Implementation

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

7.1 Base Requirements

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

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

7.1.1 CIM_AccountManagementService

At least one instance of CIM_AccountManagementService shall exist.

7.1.1.1 CIM_AccountManagementService.ElementName Constraints

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

7.1.1.1.1 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 ".*").

7.1.1.1.2 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.

7.1.2 CIM_AccountManagementCapabilities

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

7.1.2.1 CIM_AccountManagementCapabilities.ElementNameEditSupported

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

7.1.2.2 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.

7.1.3 CIM_Account

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

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

7.1.3.1 CIM_Account.UserPassword Constraints

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

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

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

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

7.1.3.1.1 UserPassword Is Clear Text

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

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

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

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

7.1.3.1.2 UserPassword Is Encrypted

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

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

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

7.1.3.2 UserID/UserPassword Usage for Authentication

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

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

7.1.3.3 UserPasswordEncoding Usage

The UserPasswordEncoding property may be used to indicate the encoding used for the UserPassword property. If the UserPasswordEncoding is Non-NULL, then the UserPassword property value shall be encoded with the encoding indicated by the UserPasswordEncoding.

When the SupportedUserPasswordEncodings[ ] property of CIM_AccountManagementCapabilities is NULL, UserPasswordEncoding may be Non-NULL.

When the SupportedUserPasswordEncodings[ ] property of CIM_AccountManagementCapabilities has no values, UserPasswordEncoding may be Non-NULL.

When the SupportedUserPasswordEncodings[ ] property of CIM_AccountManagementCapabilities contains one or more values, UserPasswordEncoding may be NULL,

When the SupportedUserPasswordEncodings[ ] property of CIM_AccountManagementCapabilities contains one or more values and UserPasswordEncoding is Non-NULL, UserPasswordEncoding shall have value set to one of the values contained in the SupportedUserPasswordEncodings[ ] property.

When the UserPassword is encrypted and encoded, then the UserPassword shall be encoded with the encoding indicated by the UserPasswordEncoding before it is encrypted.

7.1.4 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.

7.1.5 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

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

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

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

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.

If an instance of CIM_EnabledLogicalElementCapabilities representing the capabilities of an account is instantiated, then that instance shall be associated via CIM_ElementCapabilities with the instance of CIM_Account that represents that 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 configuration, the value 3 (Disabled) shall be used and the value 6 (Enabled but Offline) shall not be used.

7.3.3.5 Indicating State Management Support with CIM_EnabledLogicalElementCapabilities

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

7.3.4 CIM_Account.ElementName Constraints

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

7.3.4.1 ElementName Is Not Modifiable

The ElementNameEditSupported property shall have a value of FALSE when the implementation does not support client modification of the CIM_Account.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 ".*").

7.3.4.2 ElementName Is Modifiable

The CIM_Account.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_Account.ElementName property.

7.3.4.2.1 CIM_EnabledLogicalElementCapabilities.ElementNameEditSupported

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

7.3.4.2.2 CIM_EnabledLogicalElementCapabilities.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_Account instance.

7.3.4.2.3 CIM_EnabledLogicalElementCapabilities.ElementNameMask

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

7.3.5 Modeling Account Requirements and Capabilities

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

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

7.3.5.1 Password History Depth

The following requirements shall be met when the PasswordHistoryDepth property of an instance of CIM_AccountSettingData that is associated with the CIM_EnabledLogicalElementCapabilities 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 PasswordHistoryDepth property shall represent the maximum value that is supported for the CIM_Account.PasswordHistoryDepth property.

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

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

7.3.5.2 Password Expiration

The following requirements shall be met when the MaximumPasswordExpiration property of an instance of CIM_AccountSettingData that is associated with the CIM_EnabledLogicalElementCapabilities 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 MaximumPasswordExpiration property shall represent the maximum value expressed as an interval that is supported for the CIM_Account.PasswordExpiration property.

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

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

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

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

7.3.5.3 Complex Password Rules

The following requirements shall be met when the ComplexPasswordRulesEnforced 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 3 (Minimums), the values 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 6.2) 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 6.2) may change after the consecutive failed login attempts set by this property.

7.4 Representing a Third-Party Authenticated Principal

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_AssignedIdentity 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.

7.4.3 UserContact Creation

The ability to create UserContacts by using the CIM_AccountManagementService.CreateUserContact( ) method may be supported. This behavior is conditional. See section 8.3 for a description of the method.

7.5 Managing Account Identity Groups

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.

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

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

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

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

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

7.7 Identity Context

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

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

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

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

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

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

8 Methods

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

8.1 CIM_AccountManagementService.CreateAccount( )

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

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

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

No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

828 **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

829 ### 8.1.1 Account Template Requirements

830 This section details the requirements for the AccountTemplate parameter.

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

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

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

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

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

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

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

852 If the AccountTemplate embedded instance contains the LastLogin property, the value specified shall be
853 ignored.

854 ### 8.1.2 Account Default Values

855 This section details how default values are supplied for instances of CIM_Account that are created by
856 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 not implemented, it shall return a value of 1 (Operation unsupported).

8.2 CIM_AccountManagementService.GetAccount()

The CIM_AccountManagementService.GetAccount() method is used to retrieve a reference to an account for a specified user id.

No standard messages are defined for this method.

Table 4 – CIM_AccountManagementService.GetAccount( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 5 – CIM_AccountManagementService.GetAccount( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	UserID	String	The user ID
OUT	Account	CIM_Account REF	Reference to the Account that matches the userID

8.2.1 CIM_AccountManagementService.GetAccount( ) Conditional Support

When the OperationsSupported property of the associated instance of CIM_AccountManagementCapabilities contains the value 9 (GetAccount), the CIM_AccountManagementService.GetAccount( ) 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 9 (GetAccount), the CIM_AccountManagementService.GetAccount( ) method may be implemented; if not implemented, it shall return a value of 1 (Operation unsupported).

8.3 CIM_AccountManagementService.CreateUserContact()

The CIM_AccountManagementService.CreateUserContact( ) method is used to create instances that represent third party accounts. When the method returns a value of 0 (zero), a new instance of CIM_UserContact shall be associated with an instance of CIM_Identity (may be newly created or may have previously existed) through the CIM_AssignedIdentity association such that the values of the properties of the instance of CIM_UserContact are the values of the non-Null properties of the UserContact template instance that is specified by the UserContactTemplate parameter. The value of the UserContact parameter shall be a reference to the instance of CIM_UserContact. The instance of CIM_Identity shall be associated with the CIM_AccountManagementService through the CIM_ServiceAffectsElement association.

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

CreateUserContact( ) method return code values shall be as specified in Table 6. CreateUserContact( ) method parameters are specified in Table 7.

No standard messages are defined for this method.

Table 6 – CIM_AccountManagementService.CreateUserContact( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 7 – CIM_AccountManagementService.CreateUserContact( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	System	CIM_ComputerSystem REF	Reference to scoping system
IN, EmbeddedInstance, REQ	UserContactTemplate		Template for UserContact to create See section 8.3.1.
OUT	UserContact	CIM_UserContact REF	Reference to newly created UserContact
OUT	Identity	REF CIM_Identity	References to Identity

8.3.1 UserContact Template Requirements

This section details the requirements for the UserContactTemplate parameter.

916 If the UserContactTemplate embedded instance does not contain a non-NULL value UserID property, the
917 method shall return the value 2.

918 8.3.2 CIM_AccountManagementService.CreateUserContact( ) Conditional Support

919 When the OperationsSupported property of the associated instance of
920 CIM_AccountManagementCapabilities contains the value 5 (CreateUserContact), the
921 CIM_AccountManagementService.CreateUserContact( ) method shall be implemented and shall not
922 return a value of 1 (Unsupported). When the OperationsSupported property of the associated instance of
923 CIM_AccountManagementCapabilities does not contain the value 5 (CreateUserContact), the
924 CIM_AccountManagementService.CreateUserContact( ) method may be implemented; if not
925 implemented, it shall return a value of 1 (Operation unsupported).

926

927 8.4 CIM_AccountManagementService.CreateUserContactByIdentity()

928 The CIM_AccountManagementService.CreateUserContactByIdentity( ) method is used to create
929 instances that represent third party accounts. When the method returns a value of 0 (zero), a new
930 instance of CIM_UserContact shall be associated with the instance of CIM_Identity specified through the
931 CIM_AssignedIdentity association such that the values of the properties of the instance of
932 CIM_UserContact are the values of the non-Null properties of the UserContact template instance that is
933 specified by the UserContactTemplate parameter. The value of the UserContact parameter shall be a
934 reference to the instance of CIM_UserContact. The value of the Identity parameter shall be a reference to
935 an existing CIM_Identity instance. The instance of CIM_Identity shall be associated with the
936 CIM_AccountManagementService through the CIM_ServiceAffectsElement association.

937 When the CIM_ComputerSystem instance identified by the System parameter is not associated with the
938 CIM_AccountManagementService instance through the CIM_HostedService association, the method
939 shall return the value 2.

940 When the CIM_Identity instance identified by the Identity parameter does not exist, the method shall
941 return the value 2.

942 CreateUserContactByIdentity( ) method return code values shall be as specified in Table 6.
943 CreateUserContactByIdentity( ) method parameters are specified in Table 7.

944 No standard messages are defined for this method.

945 **Table 8 – CIM_AccountManagementService.CreateUserContactByIdentity( ) Method: Return Code**
946 **Values**

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

947 **Table 9 – CIM_AccountManagementService.CreateUserContactByIdentity( ) Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN, REQ	System	CIM_ComputerSystem REF	Reference to scoping system
IN, EmbeddedInstance, REQ	UserContactTemplate		Template for UserContact to create See section 8.3.1.

Qualifiers	Name	Type	Description/Values
IN	Identity	REF CIM_Identity	Reference to Identity
OUT	UserContact	CIM_UserContact REF	Reference to newly created UserContact

8.4.1 UserContact Template Requirements

This section details the requirements for the UserContactTemplate parameter.

If the UserContactTemplate embedded instance does not contain a value for the UserID property, the method shall return the value 2.

8.4.2 CIM_AccountManagementService.CreateUserContactByIdentity( ) Conditional Support

When the OperationsSupported property of the associated instance of CIM_AccountManagementCapabilities contains the value 6 (CreateUserContactByIdentity), the CIM_AccountManagementService.GetUserContact( ) 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 6 (CreateUserContactByIdentity), the CIM_AccountManagementService.GetUserContact( ) method may be implemented; if not implemented, it shall return a value of 1 (Operation unsupported).

8.5 CIM_AccountManagementService.GetUserContact()

The CIM_AccountManagementService.GetUserContact() method is used to retrieve a reference to a UserContact for a specified user id.

No standard messages are defined for this method.

Table 10 – CIM_AccountManagementService.GetUserContact( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 11 – CIM_AccountManagementService.GetUserContact( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	UserID	String	The User ID
OUT	UserContact	CIM_UserContact REF	Reference to the UserContact that matches the userID

8.5.1 CIM_AccountManagementService.GetUserContact( ) Conditional Support

When the OperationsSupported property of the associated instance of CIM_AccountManagementCapabilities contains the value 10 (GetUserContact), the CIM_AccountManagementService.GetUserContact( ) method shall be implemented and shall not return a value of 1 (Unsupported). When the OperationsSupported property of the associated instance of

973 CIM_AccountManagementCapabilities does not contain the value 10 (GetUserContact), the
974 CIM_AccountManagementService.GetUserContact( ) method may be implemented; if not implemented, it
975 shall return a value of 1 (Operation unsupported).

976

977 8.6 CIM_Account.RequestStateChange( )

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

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

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

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

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

988 The method shall be considered successful if the availability of the functionality upon completion of the
989 method corresponds to the desired availability indicated by the RequestedState parameter. It is not
990 necessary that an actual change in state occur for the method to be considered successful. It is sufficient
991 that the resultant state be equal to the requested state. Upon successful completion of the method, the
992 Return Value shall be 0 (zero).

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

994 RequestStateChange( ) method return code values shall be as specified in Table 12.
995 RequestStateChange( ) method parameters are specified in Table 13.

996 No standard messages are defined.

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

999 **Table 12 – 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

1000 **Table 13 – 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	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

1001 ### 8.6.1 CIM_Account.RequestStateChange( ) Conditional Support

1002 When the CIM_EnabledLogicalElementCapabilities.RequestedStatesSupported property contains at least
1003 one value, the CIM_Account.RequestStateChange( ) method shall be implemented and supported. The
1004 CIM_Account.RequestStateChange( ) method shall not return a value of 1 (Unsupported).

1005 ## 8.7 Profile Conventions for Operations

1006 For each profile class (including associations), the implementation requirements for operations, including
1007 those in the following default list, are specified in class-specific subclauses of this clause.

1008 The default list of operations is as follows:

1009 • GetInstance
1010 • Associators
1011 • AssociatorNames
1012 • References
1013 • ReferenceNames
1014 • EnumerateInstances
1015 • EnumerateInstanceNames

1016 ## 8.8 CIM_Account

1017 Table 14 lists implementation requirements for operations. If implemented, these operations shall be
1018 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 14, all operations
1019 in the default list in 8.7 shall be implemented as defined in DSP0200.

1020 NOTE: Related profiles may define additional requirements on operations for the profile class.

1021 **Table 14 – Operations: CIM_Account**

Operation	Requirement	Messages
GetInstance	Mandatory. See section 8.8.1.	None
ModifyInstance	Conditional. See section 8.8.2.	None
DeleteInstance	Conditional. See section 8.8.3.	None

1022 ### 8.8.1 CIM_Account – GetInstance Operation

1023 The following are possible behaviors and are mutually exclusive:

1024 • When the GetInstance operation is executed against an instance of CIM_Account and the
1025 underlying account has a valid password, the value of the CIM_Account.UserPassword property
1026 shall be an array of length zero to indicate that the account has a password configured and is
1027 unable or unwilling to return the value in clear text.

1028 • When the GetInstance operation is executed against an instance of CIM_Account and the
1029 underlying account does not have a valid password, the CIM_Account.UserPassword property
1030 shall be Null.

1031 ### 8.8.2 CIM_Account – ModifyInstance Operation

1032 The ModifyInstance operation shall be supported if and only if the
1033 OperationsSupported property contains the value 3 (Modify) for an instance of
1034 CIM_AccountManagementCapabilities that is associated through the
1035 CIM_ElementCapabilities association with an instance of
1036 CIM_AccountManagementService associated through CIM_ServiceAffectsElement with an instance of
1037 CIM_Identity that is associated with the instance of CIM_Account through CIM_AssignedIdentity.

1038 As described in 7.1.3.1 the UserPassword property of CIM_Account may be in clear text or be encrypted.
1039 Encrypting UserPassword may be required since the network session may not be encrypted.

1040 When the ModifyInstance operation is supported and a value is specified for the
1041 CIM_Account.UserPassword property and the CIM_Account.UserPasswordEncryptionAlgorithm property
1042 has no value or has the value 0 (None), the value of the CIM_Account.UserPassword property shall be
1043 clear text without encryption.

1044 When the ModifyInstance operation is supported and a value is specified for the
1045 CIM_Account.UserPassword property and the CIM_Account.UserPasswordEncryptionAlgorithm property
1046 has a non-zero value, the value of the CIM_Account.UserPassword property shall be encrypted in the
1047 form specified by the value of the CIM_Account.UserPasswordEncryptionAlgorithm property

1048 ### 8.8.3 CIM_Account – DeleteInstance Operation

1049 The DeleteInstance operation shall be supported if and only if the OperationsSupported property contains
1050 the value 4 (Delete) for an instance of CIM_AccountManagementCapabilities that is associated through
1051 the CIM_ElementCapabilities association with an instance of CIM_AccountManagementService
1052 associated through CIM_ServiceAffectsElement with an instance of CIM_Identity that is associated with
1053 the instance of CIM_Account through CIM_AssignedIdentity.

1054 When the associated instance of CIM_Identity is not associated with any other instances of
1055 CIM_ManagedElement through the CIM_AssignedIdentity association, the CIM_Identity instance shall be
1056 deleted.

1057 When the associated instance of CIM_EnabledLogicalElementCapabilities is not associated with any
1058 other instance of CIM_Account through the CIM_ElementCapabilities association, the instance of
1059 CIM_EnabledLogicalElementCapabilities shall be deleted.

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

8.9 CIM_EnabledLogicalElementCapabilities

All operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

8.10 CIM_AccountOnSystem

Table 15 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 15, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 15 – Operations: CIM_AccountOnSystem

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

8.11 CIM_AccountManagementCapabilities

All operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

8.12 CIM_AccountManagementService

Table 16 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 16, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 16 – Operations: CIM_AccountManagementService

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.12.1.	None

8.12.1 CIM_AccountManagementService – ModifyInstance Operation

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

8.12.1.1 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.

8.13 CIM_AccountSettingData

Table 17 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 17, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 17 – Operations: CIM_AccountSettingData

Operation	Requirement	Messages
ModifyInstance	Optional	None

8.14 CIM_AssignedIdentity

Table 18 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 18, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 18 – Operations: CIM_AssignedIdentity

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

8.15 CIM_Dependency

Table 19 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 19, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

1113 **Table 19 – Operations: CIM_Dependency**

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

1114 ## 8.16 CIM_ElementCapabilities

1115 Table 20 lists implementation requirements for operations. If implemented, these operations shall be
1116 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 20, all operations
1117 in the default list in 8.7 shall be implemented as defined in DSP0200.

1118 NOTE: Related profiles may define additional requirements on operations for the profile class.

1119 **Table 20 – Operations: CIM_ElementCapabilities**

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

1120 ## 8.17 CIM_ElementSettingData

1121 Table 21 lists implementation requirements for operations. If implemented, these operations shall be
1122 implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 21, all operations
1123 in the default list in 8.7 shall be implemented as defined in DSP0200.

1124 NOTE: Related profiles may define additional requirements on operations for the profile class.

1125 **Table 21 – Operations: CIM_ElementSettingData**

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.17.1.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

1126 ### 8.17.1 CIM_ElementSettingData – ModifyInstance

1127 The behavior of the ModifyInstance operation varies depending on the property of the association that is
1128 modified and the instances that are referenced by the association instance. The ModifyInstance operation
1129 shall not allow the IsDefault property to be modified. The ModifyInstance operation shall not allow the
1130 IsCurrent property to be modified.

1131 When the ModifyInstance operation is used to set the IsNext property to a value of 1 (Is Next), the
1132 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).

8.18 CIM_Group

All operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

8.19 CIM_HostedService

Table 22 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 22, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 22 – Operations: CIM_HostedService

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

8.20 CIM_Identity

All operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

8.21 CIM_IdentityContext

Table 23 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 23, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 23 – Operations: CIM_IdentityContext

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

8.22 CIM_MemberOfCollection

Table 24 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 24, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 24 – Operations: CIM_MemberOfCollection

Operation	Requirement	Messages
CreateInstance	Optional. See section 8.22.1.	None
DeleteInstance	Optional. See section 8.22.2.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.22.1 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.

8.22.2 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.

8.23 CIM_OwningCollectionElement

Table 25 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 25, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 25 – Operations: CIM_OwningCollectionElement

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

8.24 CIM_ServiceAffectsElement

Table 26 lists implementation requirements for operations. If implemented, these operations shall be

implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 26, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 26 – Operations: CIM_ServiceAffectsElement

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

8.25 CIM_SettingsDefineCapabilities

Table 27 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 27, all operations in the default list in 8.7 shall be implemented as defined in DSP0200.

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

Table 27 – Operations: CIM_SettingsDefineCapabilities

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

8.26 CIM_UserContact

All operations in the default list in 8.7 shall be implemented as defined in DSP0200.

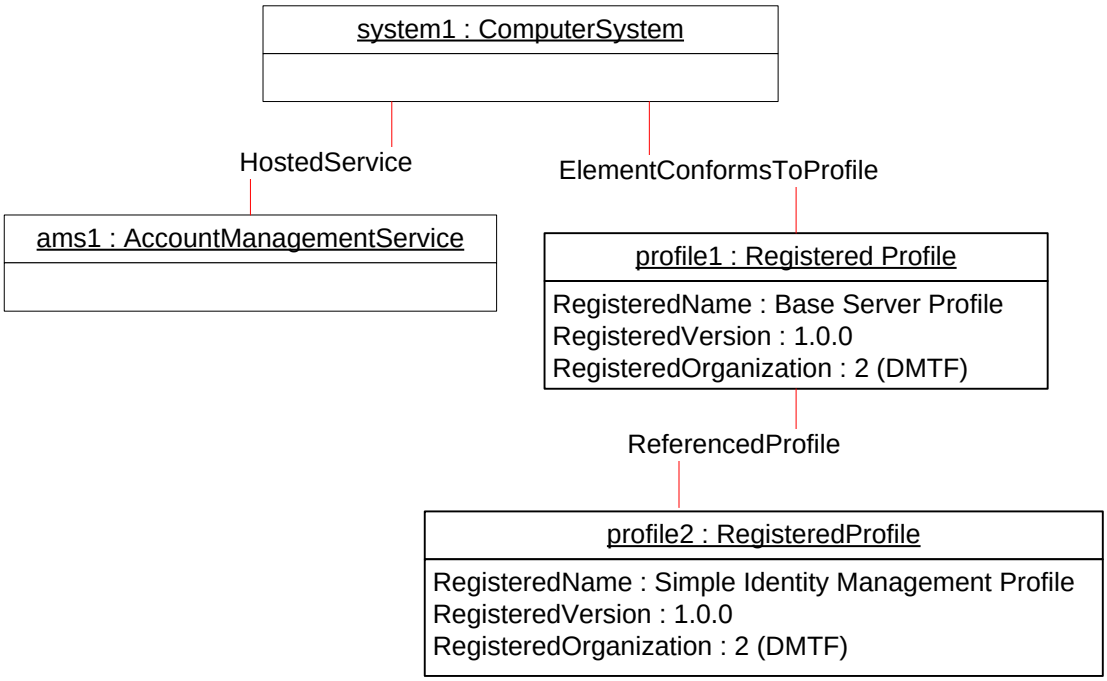
NOTE: Related profiles may define additional requirements on operations for the profile class.

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.



1203

1204 **Figure 2 – Profile Registration**

1205 Figure 3 shows a system that supports management of local accounts for authentication and
1206 authorization. The modeled system supports creation, modification, and deletion of accounts. Privilege
1207 management is performed through assignment to Roles.

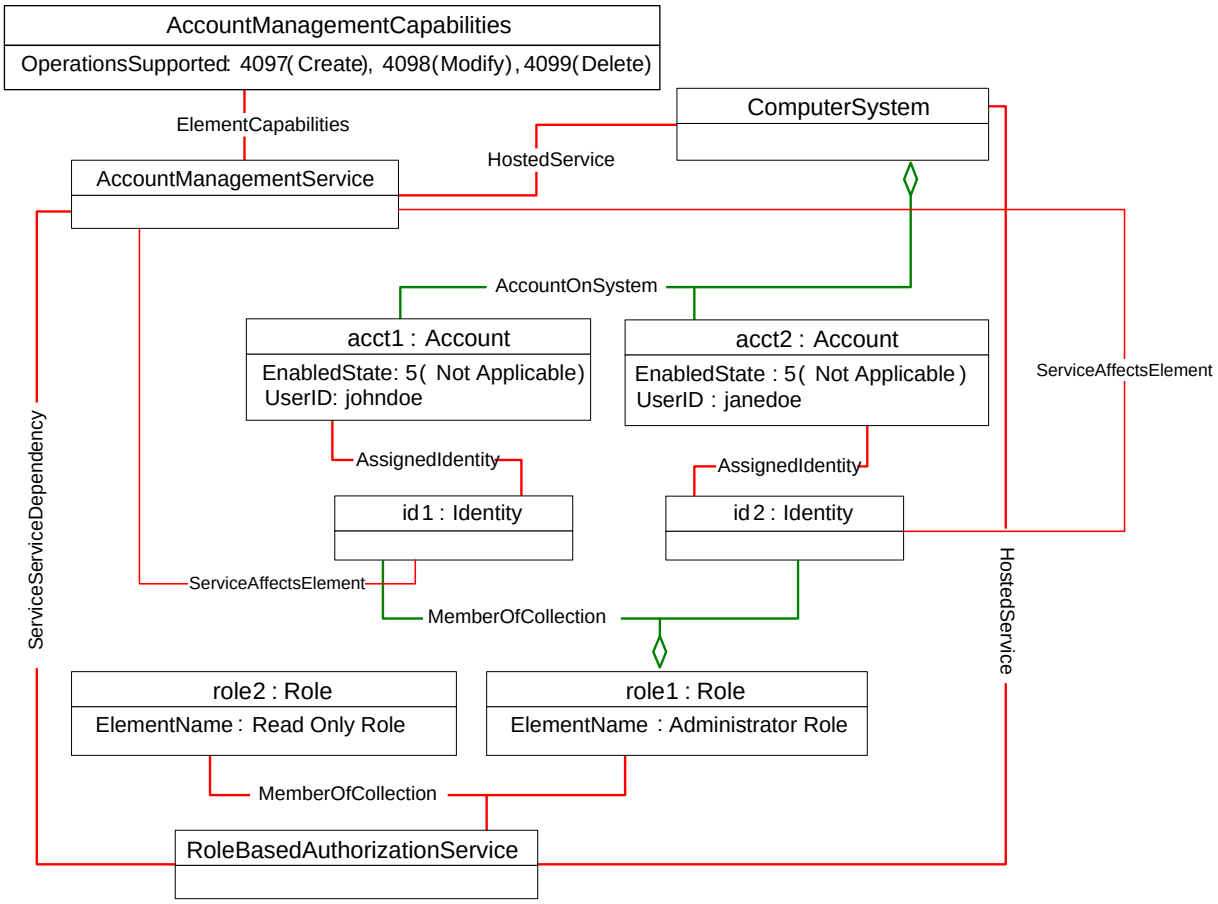
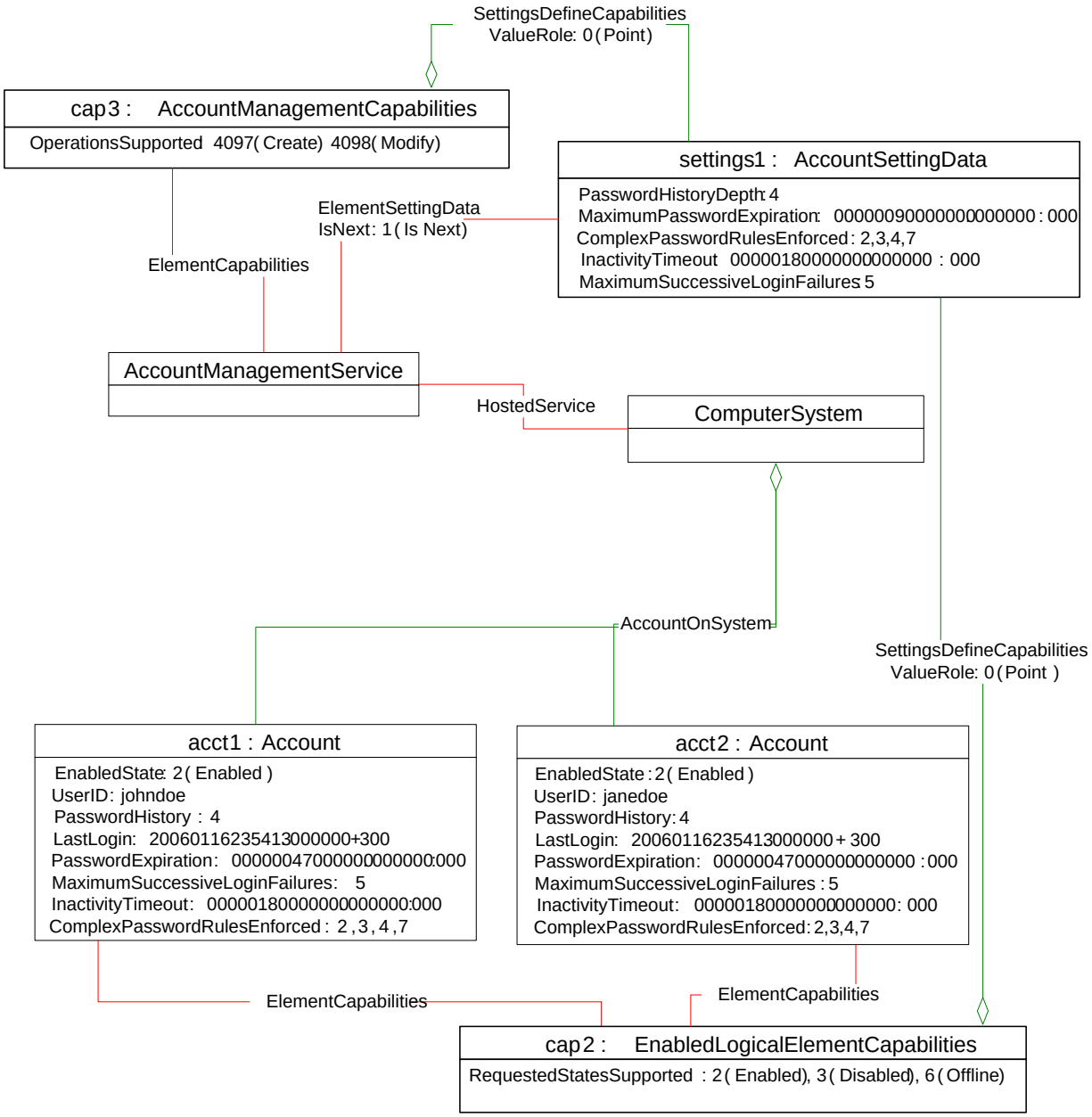


Figure 3 – Basic System Accounts

Figure 4 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, have the password changed every 90 days, enter a locked-out state after 180 days of inactivity, and enter a locked-out state after five successive failed login attempts. 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 requirements are indicated by the CIM_SettingsDefineCapabilities association between settings1 and cap3.

acct1 and acct2 operate under the same password constraints as new accounts created through the CIM_AccountManagementService. This behavior is indicated by the CIM_SettingsDefineCapabilities association between cap2 and settings1. 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.



1224

Figure 4 –

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.

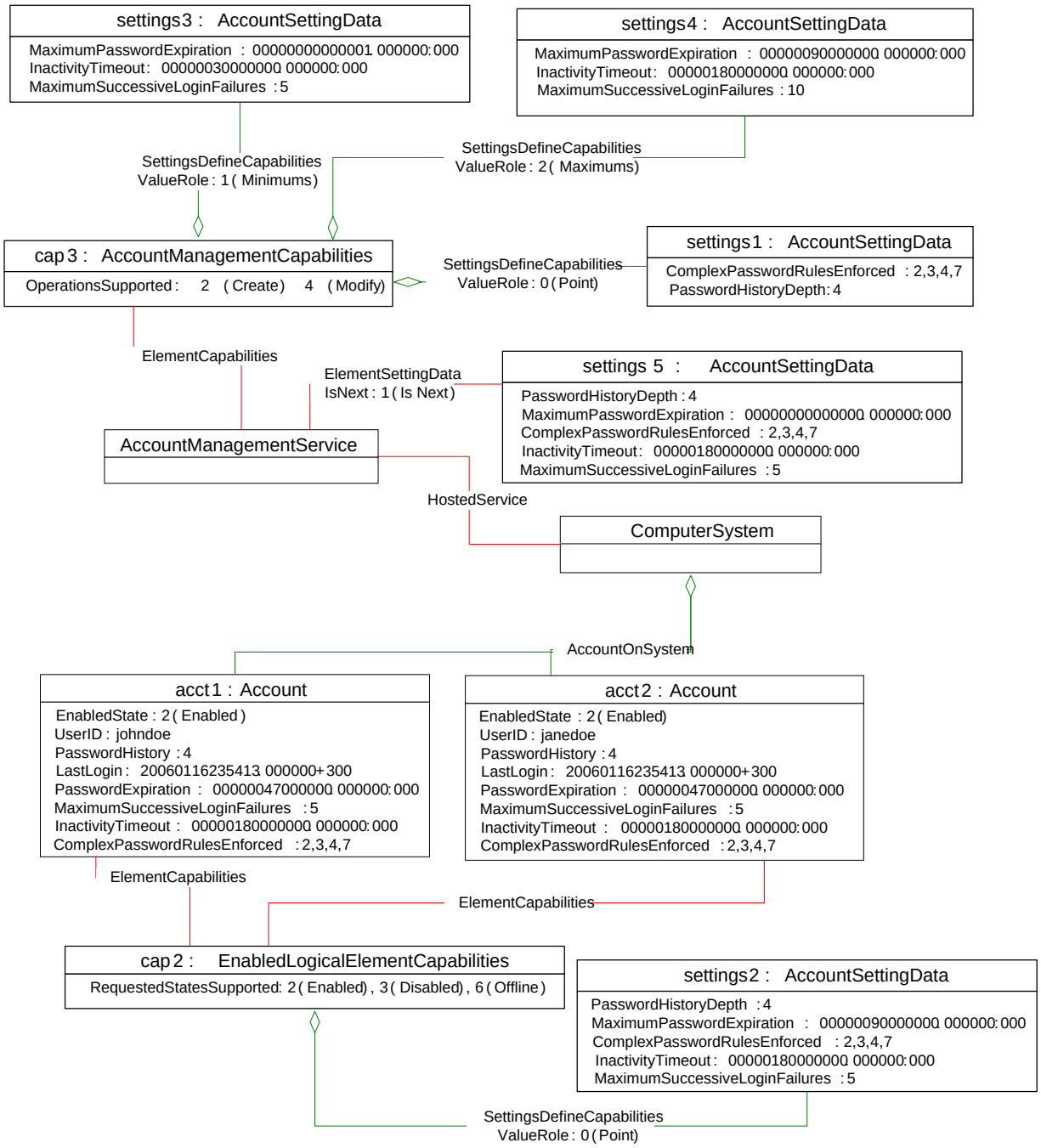


Figure 5 – Account Capabilities with Ranges

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.

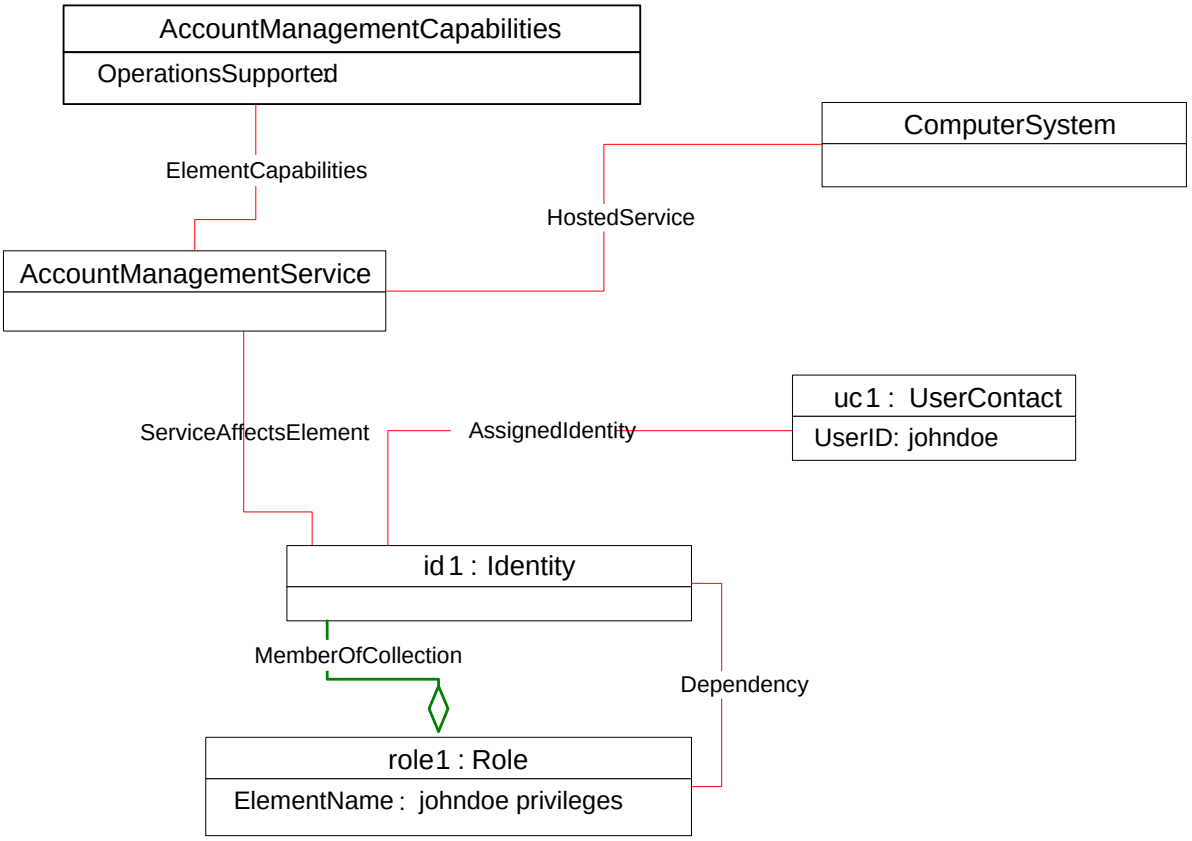
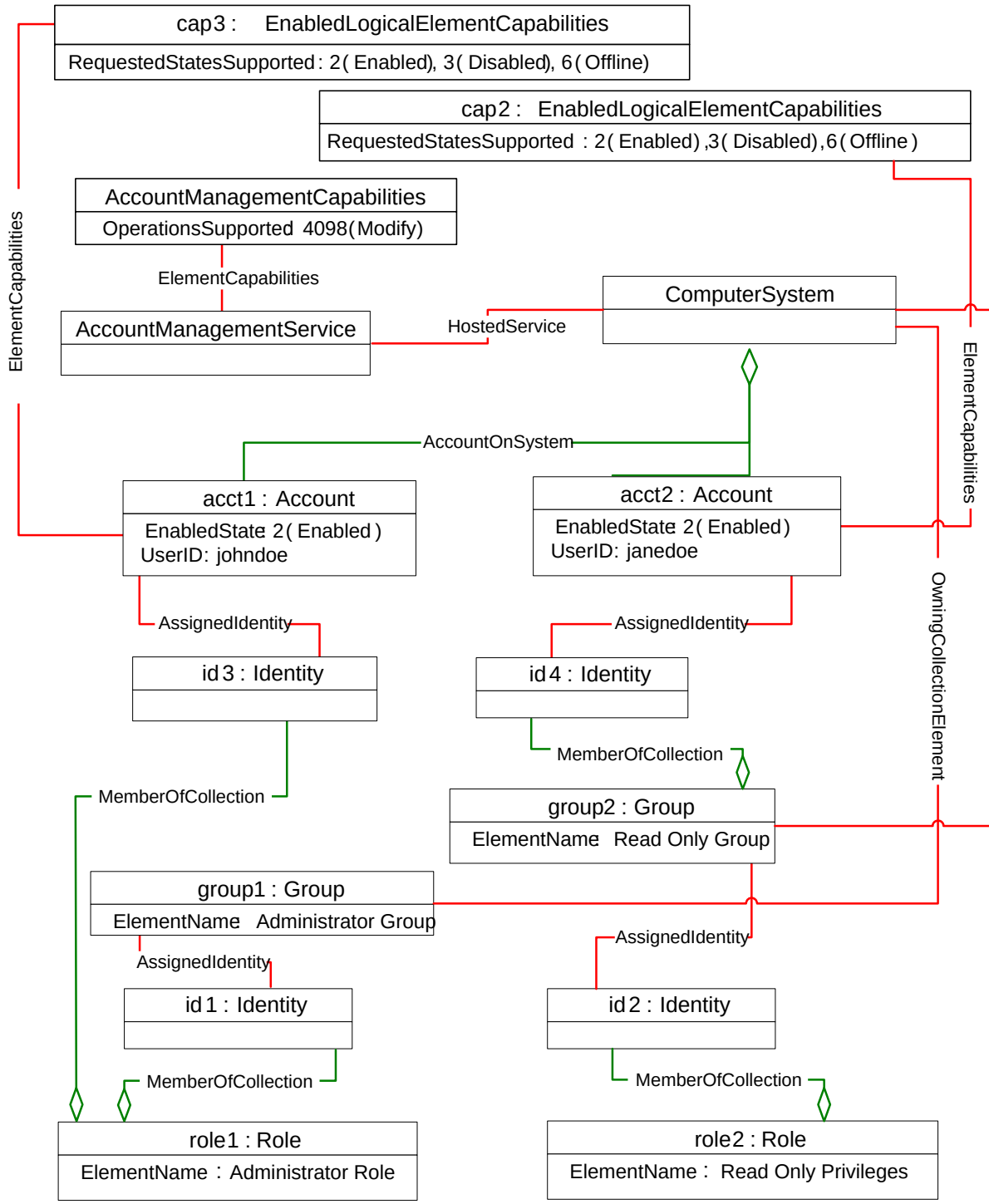


Figure 6 – Third-Party Authenticated User

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.

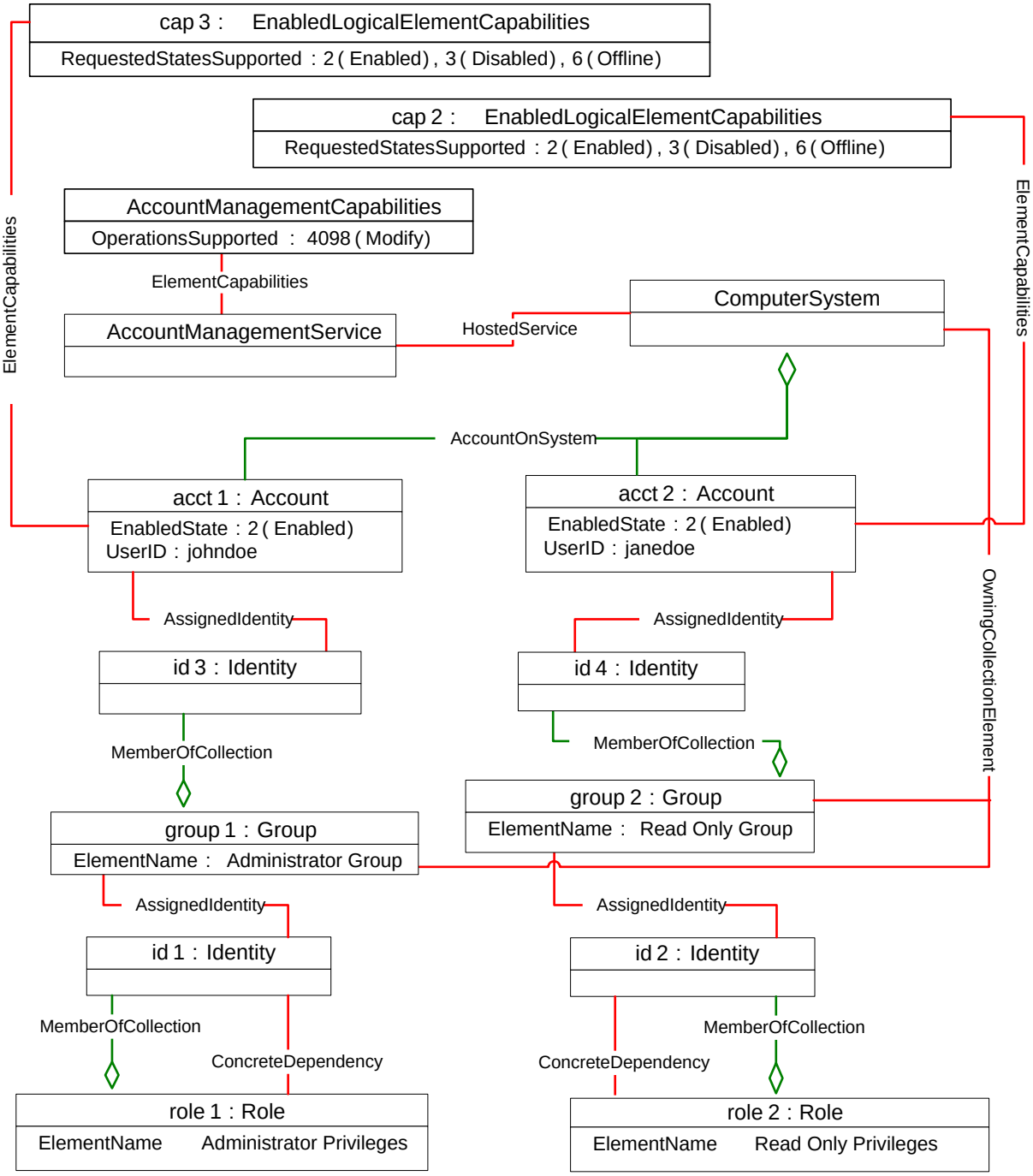


1262

1263 **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.



1280

1281 **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.

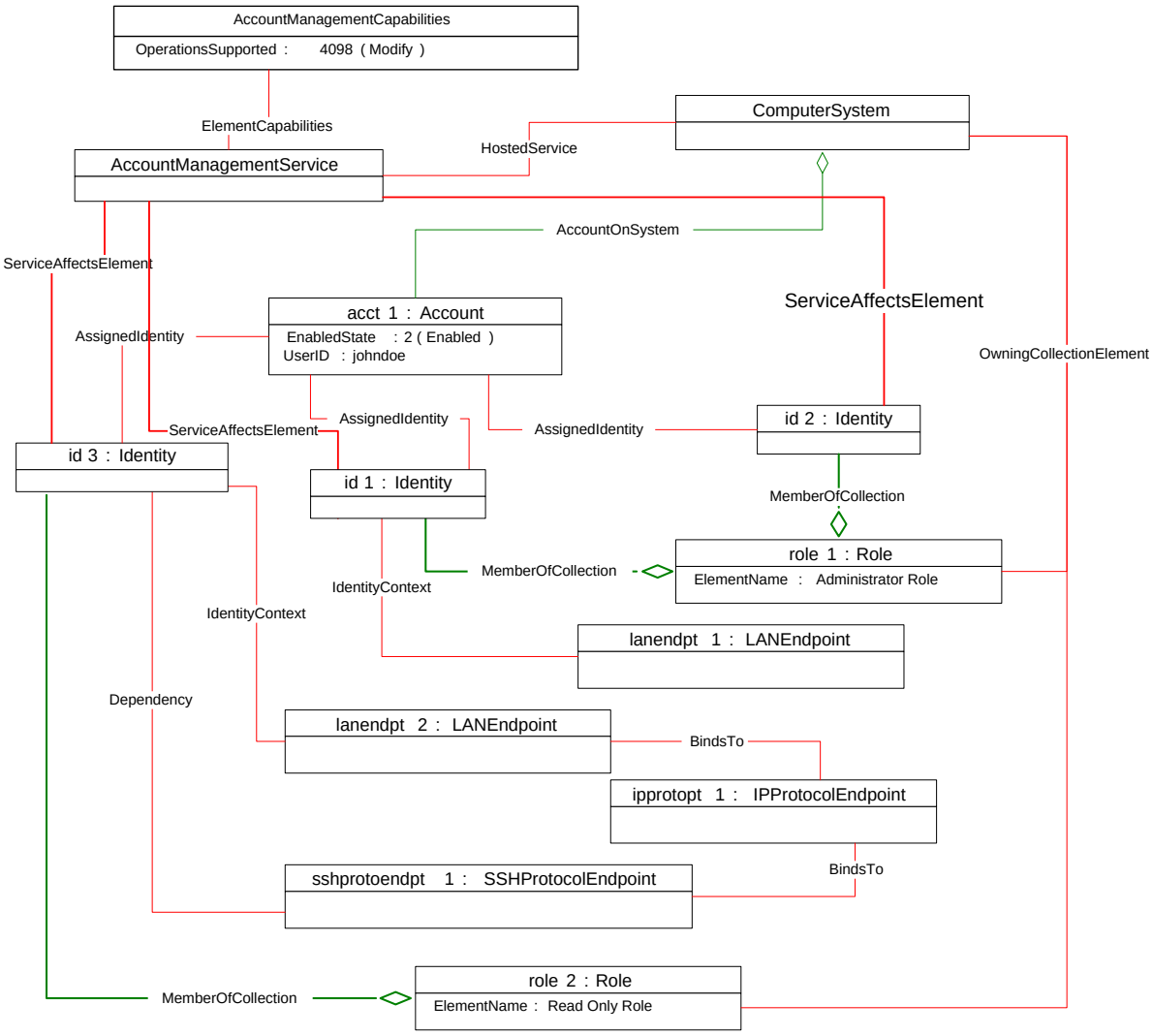


Figure 9 – Access Ingress Point and Identity Context

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.
2) 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.
2) 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.
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 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:

4) Determine if account creation is supported as follows:
   a) 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.
   b) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
   c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

      If the value 2 (Create) is contained in the array, account creation is supported.
5) Create a template instance of CIM_Account.
6) 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).
2) 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:
   a) 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
   b) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
   c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

   If the value 4 (Delete) is contained in the array, account deletion is supported.
2) 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:
   a) 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
   b) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
   c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

   If the value 3 (Modify) is contained in the array, account modification is supported.
2) Invoke the GetInstance operation against the target instance of CIM_Account
3) Modify the UserPassword property.
4) 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.
2) If an instance is found, query the RequestedStatesSupported property to determine if it contains the value 3 (Disabled).
3) 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.
2) If an instance is found, query the RequestedStatesSupported property to determine if it contains the value 6 (Enabled but Offline).
3) 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.

9.14 Create a UserContact

A client can create a UserContact on a system as follows:

1) Determine if usercontact creation is supported as follows:
   a) 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.
   b) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
   c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

      If the value 5 (CreateUserContact) is contained in the array, usercontact creation is supported.
2) Create a template instance of CIM_UserContact.
3) Invoke the CIM_AccountManagementService.CreateUserContact( ) method, specifying the template instance.

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

9.15 Get UserContact

A client can retrieve the CIM_UserContact instance that represents a UserID on a system as follows:

1) Determine if the GetUserContact method is supported as follows:
   a) 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.
   b) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
   c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

      If the value 10 (GetUserContact) is contained in the array, the method is supported.
2) Invoke the CIM_AccountManagementService.GetUserContact( ) method, specifying the UserID.

9.16 Get Account

A client can retrieve the CIM_Account instance that represents a UserID on a system as follows:

1) Determine if the GetAccount method is supported as follows:
   a) 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.
   b) Find an instance of CIM_AccountManagementCapabilities that is associated with the CIM_AccountManagementService instance through the CIM_ElementCapabilities association.
   c) Query the value of the CIM_AccountManagementCapabilities.OperationsSupported property.

      If the value 9 (GetAccount) is contained in the array, the method is supported.
2) Invoke the CIM_AccountManagementService.GetAccount( ) method, specifying the UserID.

1454

1455 10 CIM Elements

1456 Table 28 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
1457 implemented as described in Table 28. Sections 7 ("Implementation") and 8 ("Methods") may impose
1458 additional requirements on these elements.

1459

1460 **Table 28 – CIM Elements: *Simple Identity Management Profile***

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

10.1 CIM_Account

Table 29 details the requirements for instances of CIM_Account.

Table 29 – Class: CIM_Account

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
Name	Mandatory	Key
UserID	Mandatory	(pattern ".*")
UserPassword	Mandatory	(pattern ".*")
OrganizationName	Mandatory	(pattern ".*")
ElementName	Mandatory	See section 7.3.4.1.
UserPasswordEncryptionAlgorithm	Optional	See section 7.1.3.1.
OtherUserPasswordEncryptionAlgorithm	Conditional	Mandatory when UserPasswordEncryptionAlgorithm is 1 (Other).
PasswordHistoryDepth	Optional	See section 7.3.5.1.
PasswordExpiration	Optional	See section 7.3.5.2.
ComplexPasswordRulesEnforced	Optional	See section 7.3.5.3.
InactivityTimeout	Optional	See section 7.3.5.4.
MaximumSuccessiveLoginFailures	Optional	See section 7.3.5.5.
RequestedState	Mandatory	See section 7.3.3.3.
EnabledState	Mandatory	See section 7.3.3.4.
UserPasswordEncoding	Optional	See section 7.1.3.3.
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 30 details the requirements for instances of CIM_AccountManagementCapabilities.

Table 30 – 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
MaximumAccountsSupported	Optional	None
SupportedUserPasswordEncodings	Optional	See section 7.1.3.3.
SupportedUserPasswordEncryptionAlgorithms[ ]	Optional	See section 7.1.2.

10.3 CIM_AccountManagementService

Table 31 details the requirements for instances of CIM_AccountManagementService.

Table 31 – 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.
GetAccount()	Conditional	See section 8.2.
CreateUserContact()	Conditional	See section 8.3.
CreateUserContactByIdentity()	Optional	See Section
GetUserContact()	Conditional	See section 8.5.

10.4 CIM_AccountOnSystem

Table 32 details the requirements for instances of CIM_AccountOnSystem.

Table 32 – 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 33 details the requirements for instances of CIM_AccountSettingData.

Table 33 – Class: CIM_AccountSettingData

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

10.6 CIM_AssignedIdentity (CIM_Account)

Table 34 details the requirements for instances of CIM_AssignedIdentity.

Table 34 – Class: CIM_AssignedIdentity (CIM_Account)

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

10.7 CIM_AssignedIdentity (Group)

Table 35 details the requirements for instances of CIM_AssignedIdentity.

Table 35 – 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 36 details the requirements for instances of CIM_AssignedIdentity.

Table 36 – 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 37 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 37 – 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 38 details the requirements for instances of CIM_ElementCapabilities.

Table 38 – 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 39 details the requirements for instances of CIM_ElementCapabilities.

Table 39 – 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 40 details the requirements for instances of CIM_ElementSettingData.

Table 40 – 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 41 details the requirements for instances of CIM_EnabledLogicalElementCapabilities.

Table 41 – 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 42 details the requirements for instances of CIM_Group.

Table 42 – Class: CIM_Group

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

10.15 CIM_HostedService

Table 43 details the requirements for instances of CIM_HostedService.

Table 43 – 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 44 details the requirements for instances of CIM_Identity.

Table 44 – Class: CIM_Identity

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

10.17 CIM_IdentityContext

Table 45 details the requirements for instances of CIM_IdentityContext.

Table 45 – 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 46 details the requirements for instances of CIM_MemberOfCollection when it is used to associate instances of CIM_Identity with instances of CIM_Group.

Table 46 – 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..*

10.19 CIM_OwningCollectionElement

Table 47 details the requirements for instances of CIM_OwningCollectionElement.

Table 47 – 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 *

10.20 CIM_ServiceAffectsElement

Table 48 details the requirements for instances of CIM_ServiceAffectsElement.

Table 48 – 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)

10.21 CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)

Table 49 details the requirements for instances of CIM_SettingsDefineCapabilities when it is used to associate an instance of CIM_AccountSettingData with an instance of CIM_AccountManagementCapabilities. The value of the PropertyPolicy property is fixed at 0 (Independent), which indicates that the value of each property on the referenced CIM_AccountSettingData instances is independent of the values of the other properties. The ValueRole[] property is fixed at the value 3 (Supported), which indicates that the value of each property on a referenced instance of CIM_AccountSettingData represents an inclusive constraint.

Table 49 – Class: CIM_SettingsDefineCapabilities (CIM_AccountManagementCapabilities)

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

10.22 CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)

Table 50 details the requirements for instances of CIM_SettingsDefineCapabilities when it is used to associate an instance of CIM_AccountSettingData with an instance of CIM_EnabledLogicalElementCapabilities. The value of the PropertyPolicy property is fixed at 0 (Independent), which indicates that the value of each property on the referenced CIM_AccountSettingData instances is independent of the values of the other properties. The ValueRole[] property is fixed at the value 3 (Supported), which indicates that the value of each property on a referenced instance of CIM_AccountSettingData represents an inclusive constraint.

1548 **Table 50 – Class: CIM_SettingsDefineCapabilities (CIM_EnabledLogicalElementCapabilities)**

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

1549 ## 10.23 CIM_UserContact

1550 Table 51 details the requirements for instances of CIM_UserContact.

1551 **Table 51 – Class: CIM_UserContact**

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

1552 ## 10.24 CIM_RegisteredProfile

1553 CIM_RegisteredProfile identifies the *Simple Identity Management Profile* The CIM_RegisteredProfile
1554 class is defined by the DSP1033. With the exception of the mandatory values specified for the properties
1555 in Table 52, the behavior of the CIM_RegisteredProfile instance is defined by the DSP1033.

1556 **Table 52 – Class: CIM_RegisteredProfile**

Elements	Requirement	Notes
RegisteredName	Mandatory	Matches "Simple Identity Management"
RegisteredVersion	Mandatory	Matches "1.1.0"
RegisteredOrganization	Mandatory	Matches 2 ("DMTF")

1557

ANNEX A
(informative)

Change Log

Version	Date	Description
1.0.0	07/23/2008	
1.0.1	06/17/2009	DMTF Standard
1.1.0	12/13/2012	DMTF Standard