Document Number: DSP1005

Date: 2009-06-17

Version: 1.0.0

Command Line Protocol Service Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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 *Command Line Protocol Service Profile* (DSP1005) was prepared by the Physical Platform Profiles Working Group and the Server Management 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.

Contributors:

- Aaron Merkin

Participants from the DMTF Server Management Working Group:

- Jon Hass – Dell
- Khachatur Papanyan – Dell
- Enoch Suen – Dell
- Jeff Hilland – HP
- Christina Shaw – HP
- Perry Vincent – Intel
- John Leung – 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 a CLP service, its associated configuration information, and any active connections.

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.

Command Line Protocol Service Profile

1 Scope

The *Command Line Protocol Service Profile* extends the management capability of referencing profiles by adding the capability to represent a CLP service and its associated sessions. A CLP service is logical entity that provides management access through the *Server Management Command Line Profile* defined in the *Server Management Command Line Protocol Specification*.

2 Normative References

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

2.1 Approved References

DMTF DSP0004, *CIM Infrastructure Specification 2.3,*
http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

DMTF DSP0200, *CIM Operations over HTTP 1.2,*
http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

DMTF DSP0214, *Server Management Command Line Protocol Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0214_1.0.pdf

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

DMTF DSP1007, *SM CLP Admin Domain Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1007_1.0.pdf

DMTF DSP1016, *Telnet Service Profile 1.0,*
http://www.dmtf.org/standards/published_documents/DSP1016_1.0.pdf

DMTF DSP1017, *SSH Service Profile 1.0,*
http://www.dmtf.org/standards/published_documents/DSP1017_1.0.pdf

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

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

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

2.2 Other References

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

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. For the purposes of this document, the terms and definitions given in *Profile Registration Profile* and *Management Profile Specification Usage Guide* also apply.

3.1

can

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

3.2

cannot

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

3.3

conditional

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

3.4

mandatory

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

3.5

may

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

3.6

need not

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

3.7

optional

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

3.8

referencing profile

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

3.9

shall

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

3.10

shall not

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

3.11

should

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

3.12
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

3.13
unspecified
indicates that this profile does not define any constraints for the referenced CIM element or operation

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1
CIM
Common Information Model

4.2
CLP
Command Line Protocol

4.3
IP
Internet Protocol

4.4
TCP
Transmission Control Protocol

5 Synopsis

Profile Name: Command Line Protocol Service

Version: 1.0.0

Organization: DMTF

CIM Schema Version: 2.22

Central Class: CIM_ProtocolService

Scoping Class: CIM_ComputerSystem

The *Command Line Protocol Service Profile* extends the management capability of referencing profiles by adding the capability to represent a CLP service in a managed system. This profile includes a specification of the CLP service, its associated configuration, and any active sessions.

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

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Simple Identity Management	DMTF	1.0	Mandatory	See section 7.5.
Profile Registration	DMTF	1.0	Mandatory	None
Role Based Authorization	DMTF	1.0	Mandatory	See section 7.5.
SM CLP Admin Domain	DMTF	1.0	Mandatory	See section 7.4.
SSH Service	DMTF	1.0	Optional	See section 7.3.
Telnet Service	DMTF	1.0	Optional	See section 7.3.

The Central Class for the *Command Line Protocol Service Profile* shall be the CIM_ProtocolService class. The Central Instance of the *Command Line Protocol Service Profile* shall be an instance of CIM_ProtocolService. The Scoping Class for the *Command Line Protocol Service Profile* shall be CIM_ComputerSystem. The Scoping Instance of the *Command Line Protocol Service Profile* shall be the instance of CIM_ComputerSystem to which the Central Instance is associated through an instance of the CIM_HostedService association.

6 Description

The *Command Line Protocol Service Profile* describes a CLP service, its associated configuration, and active sessions. A CLP Service is a conceptual entity that supports the protocol specified in *Server Management Command Line Protocol Specification* (DSP0214). DSP0214 identifies numerous requirements for configuration and management functionality of the CLP service itself. The *Command Line Protocol Service Profile* specifies how to satisfy these requirements through representation and manipulation of CIM elements.

Figure 1 represents the class schema for the *Command Line Protocol Service Profile*.

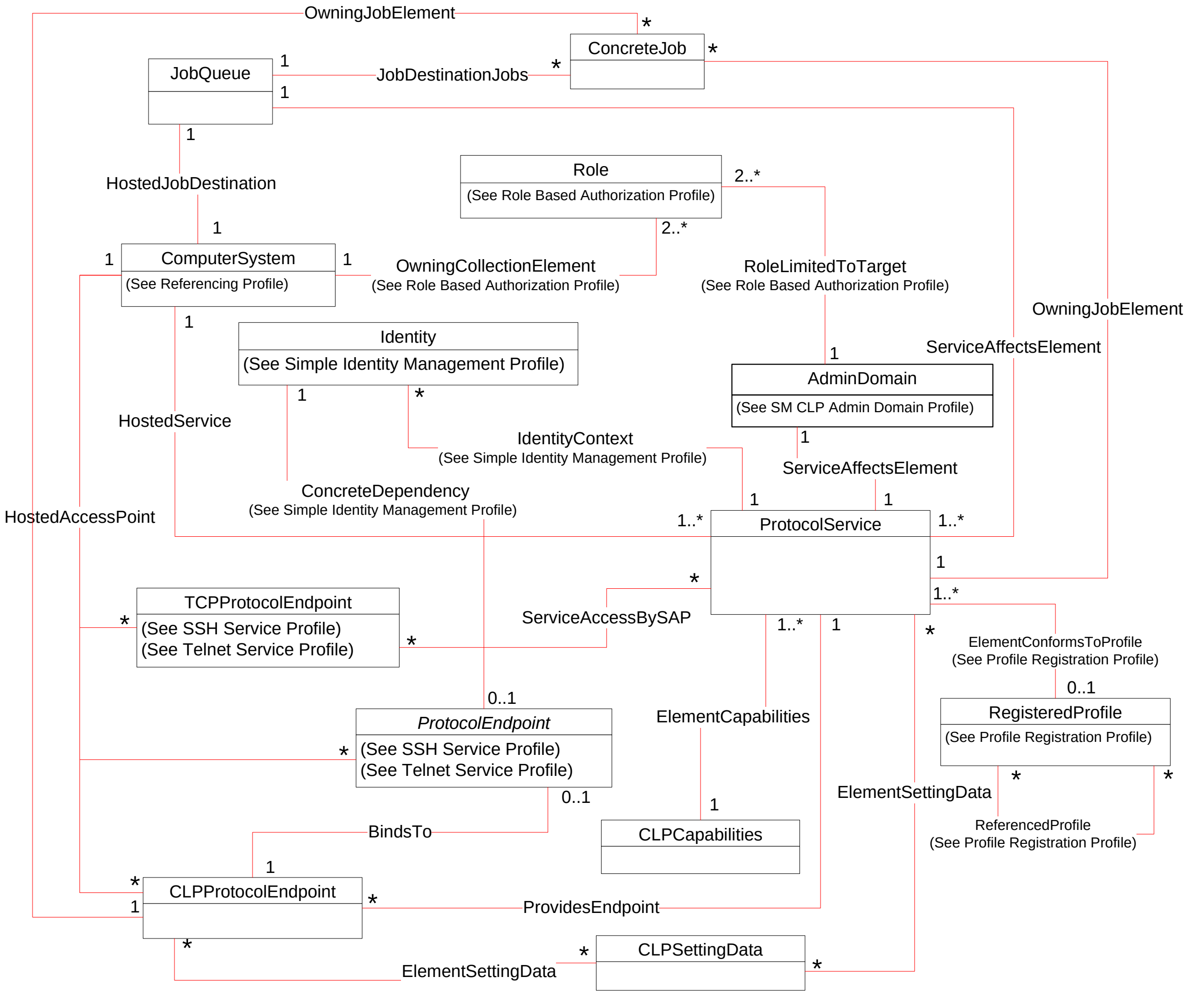


Figure 1 – Command Line Protocol Service Profile: Class Diagram

The *Command Line Protocol Service Profile* extends the management capability of referencing profiles by adding the capability to represent a CLP service hosted on a managed system. Functionality within the scope of this profile includes:

- representation of the CLP service
- representation of active CLP sessions at the CLP server
- configuration of the CLP service
- configuration of the CLP sessions from the CLP server

Functionality explicitly excluded from the scope of this profile includes modeling of the CLP session at the CLP client.

This profile represents the capabilities of the CLP service, the current configuration of the CLP service, active sessions, and the default settings when new sessions are activated. The CLP service is represented by an instance of CIM_ProtocolService. The capabilities of the CLP service are represented by an instance of CIM_CLPCapabilities. The current configuration of the CLP service is modeled with the properties from the instance of CIM_ProtocolService. Each active session with the CLP service is represented by an instance of CIM_CLPProtocolEndpoint. The current configuration of an active session is reflected in the values of the properties from the CIM_CLPProtocolEndpoint. CIM_CLPSettingData represents a complete configuration that a CLP session could have. When associated with CIM_ProtocolService, an instance of CIM_CLPSettingData contains a configuration that could be in effect for a CLP session when it is first established. When associated with a CIM_CLPProtocolEndpoint, an instance of CIM_CLPSettingData contains the configuration that was in effect when the CLP session was established.

DSP0214 specifies requirements for the management of CLP operations. CLP operations and the job queue are modeled using CIM_ConcreteJob and CIM_JobQueue respectively. See section 7.6.

6.1 CLP Session Lifecycle

When a CLP session is established with the CLP service, an instance of CIM_CLPProtocolEndpoint is created. The CIM_CLPProtocolEndpoint instance exists for the duration of the CLP session that it represents. When the CLP session is ended, the CIM_CLPProtocolEndpoint will be removed. When the CIM_CLPProtocolEndpoint is explicitly deleted through an intrinsic DeleteInstance operation, the CLP session is ended.

7 Implementation Requirements

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

7.1 Representing a CLP Service

An instance of CIM_ProtocolService shall represent the CLP service being modeled.

7.1.1 CIM_ProtocolService.Protocol

The Protocol property of the CIM_ProtocolService instance shall have a value of 4 (CLP).

7.1.2 CLP Service Capabilities

An instance of CIM_CLPCapabilities shall be associated with the CIM_ProtocolService instance through an instance of CIM_ElementCapabilities. This instance of CIM_CLPCapabilities shall represent the capabilities of the CLP service.

7.1.2.1 Maximum Number of Sessions Supported

A value of zero for the CIM_CLPCapabilities.MaxConnections property shall indicate that the CLP service does not enforce a limit on the number of concurrent sessions supported. The underlying transport protocol may enforce a limit on the number of concurrent session which would in turn result in a restriction in the number of concurrent CLP sessions. When the optional behavior of modeling the underlying transport is implemented (see section 7.3), the CIM_ProtocolServiceCapabilities.MaxConnections property for the CIM_ProtocolService instance representing the transport will indicate the maximum number of concurrent sessions for the transport.

7.1.3 Managing the CLP Service's State

This section describes the usage of the RequestedState and EnabledState properties to represent the state of an instance of CIM_ProtocolService.

7.1.3.1 State Management Supported

Exactly one instance of CIM_CLPCapabilities shall be associated with an instance of CIM_ProtocolService, which indicates support for managing the state of the CLP service.

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

7.1.3.2 CIM_ProtocolService.RequestStateChange( ) Supported

When the CIM_CLPCapabilities.RequestedStatesSupported property contains at least one value, the CIM_ProtocolService.RequestStateChange( ) method shall be implemented and supported. The CIM_ProtocolService.RequestStateChange( ) method shall not return a value of 1 (Unspecified).

7.1.3.3 CIM_ProtocolService.RequestedState

When state management is supported, the RequestedState property shall be supported. When state management is Unspecified, the RequestedState property may be supported.

Upon successful invocation of the CIM_ProtocolService.RequestStateChange( ) method, 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_CLPCapabilities contains one or more values, the RequestedState property shall have one of the values specified or 5 (No Change). When the RequestedStatesProperty of the associated instance of CIM_EnabledLogicalElementCapabilities does not contain any values, the RequestedState property shall have the value of 12 (Not Applicable).

7.1.3.4 EnabledState

When the RequestedState parameter has a value of 2 (Enabled) or 3 (Disabled), upon successful completion of the CIM_ProtocolService.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), or 5 (Not Applicable).

7.1.3.5 Indicating State Management Support with CIM_CLPCapabilities

When state management is supported, the RequestedStatesSupported property of the CIM_CLPCapabilities instance associated with the CIM_ProtocolService instance via 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 11 (Reset).

7.1.4 CIM_ProtocolService ElementName Constraints

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

7.1.4.1 ElementName Is Not Modifiable

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 ".*").

7.1.4.2 ElementName Is Modifiable

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

7.1.4.2.1 CIM_CLPCapabilities.ElementNameEditSupported

This property shall have a value of TRUE when the implementation supports client modification of the CIM_ProtocolService.ElementName property.

7.1.4.2.2 CIM_CLPCapabilities.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_ProtocolService instance.

7.2 Representing a CLP Session

Each active session with the CLP service shall be represented with an instance of CIM_CLPProtocolEndpoint.

7.2.1 Relationship with Service

An instance of CIM_ProvidesEndpoint shall associate the CIM_ProtocolService with the CIM_CLPProtocolEndpoint.

7.2.2 Specification Default Configuration

DSP0214 defines default values for each session attribute that is required to be maintained. This is the specification default configuration and shall be represented by an instance of CIM_CLPSettingData implemented as specified in section 10.4. This instance of CIM_CLPSettingData shall be associated with the Central Instance through the CIM_ElementSettingData association where the IsDefault property of the CIM_ElementSettingData instance has the value 1 (Is Default).

7.2.3 Session Default Configuration

When a CLP session is created, it will have an initial configuration. Implementations can indicate to clients the configuration that will be assigned to a session. An implementation can also indicate to clients the configuration that an active session had when the session was first established. This specification does not identify requirements for representing the configuration that will be utilized when a specific user establishes a session.

7.2.3.1 Configuration that Will Be Assigned (Optional)

The same initial configuration may be assigned for all CLP sessions spawned. This is optional behavior. When the implementation assigns the same initial configuration for all CLP sessions, the configuration that a session will have when it is established shall be represented by an instance of CIM_CLPSettingData associated with the CIM_ProtocolService through an instance of CIM_ElementSettingData where the IsNext property of the CIM_ElementSettingData instance has a value of 1 (Is Next).

7.2.3.2 Initial Configuration of a Session (Optional)

The initial configuration of a session may be modeled. This is optional behavior. When the configuration that a session had when it was established is modeled, it shall be represented by an instance of CIM_CLPSettingData associated with the CIM_CLPProtocolEndpoint through an instance of CIM_ElementSettingData where the IsCurrent property of the CIM_ElementSettingData instance has a value of 1 (Is Current).

It is not necessary that there be a discrete copy of CIM_CLPSettingData for each active session. It is only necessary that the CIM_CLPSettingData associated with the CIM_CLPProtocolEndpoint accurately reflect the initial configuration of the session.

7.3 Relationship with Transport Services (Optional)

DSP0214 indicates support for accessing the CLP using either SSH or Telnet as the transport protocol. The ability to access the CLP through SSH or Telnet may be modeled. When the ability to access the CLP over SSH is modeled, the *SSH Service Profile* shall be implemented. When the ability to access the CLP over Telnet is modeled, the *Telnet Service Profile* shall be implemented. When the ability to access the CLP through a transport protocol is modeled, the behavior in the following sections shall be implemented.

7.3.1 Access via SSH

A CLP implementation may be accessible via SSH. When the SSH service underlying the CLP service is modeled the requirements in this section shall be met.

The *SSH Service Profile* shall be implemented. The optional behavior specified in section 7.1.5.1.1 of the *SSH Service Profile* should be implemented. After a CLP session has been established through an SSH session, there shall be an instance of CIM_BindsTo that associates the CIM_CLPProtocolEndpoint representing the CLP session with the CIM_SSHProtocolEndpoint representing the SSH session.

7.3.2 Access via Telnet

A CLP implementation may be accessible via Telnet. When the Telnet service underlying the CLP service is modeled the requirements in this section shall be met.

The *Telnet Service Profile* shall be implemented. The optional behavior specified in section 7.2.2.1 of *Telnet Service Profile* should be implemented. After a CLP session has been established through a Telnet session, there shall be an instance of CIM_BindsTo that associates the CIM_CLPProtocolEndpoint representing the CLP session with the CIM_TelnetProtocolEndpoint representing the Telnet session.

7.3.3 Port(s) Through Which the CLP Can Be Accessed

When the optional behavior specified in section 7.1.5.1.1 of *SSH Service Profile* or the optional behavior specified in section 7.2.2.1 of *Telnet Service Profile* is implemented, there will be at least one instance of CIM_TCPProtocolEndpoint.

For each instance of CIM_TCPProtocolEndpoint, if establishing a connection to the transport protocol through the port represented by the instance of CIM_TCPProtocolEndpoint will establish, or enable the establishment of, a CLP session, the Central Instance shall be associated to the instance of CIM_TCPProtocolEndpoint through an instance of CIM_ServiceAccessBySAP.

7.4 SM CLP Admin Domain

The *SM CLP Admin Domain Profile* shall be implemented. There shall be an instance of CIM_ServiceAffectsElement that associates the Central Instance of this profile with the Central Instance of the *SM CLP Admin Domain Profile.*

7.5 CLP User Account Management (Mandatory)

This section describes the requirements for representing CLP groups and authorization. The *Simple Identify Management Profile* and the *Role Based Authorization Profile* shall be implemented.

7.5.1 Constraining Role Usage

This section details constraints on associations defined in the *Role Based Authorization Profile* that are used to indicate the scope of an instance of CIM_Role.

7.5.1.1 Role Scope

Each instance of CIM_Role implemented as defined in sections 7.5.2.1, 7.5.3.1, and 7.5.4.1 shall be associated with the CIM_AdminDomain instance specified in section 7.4 through the CIM_RoleLimitedToTarget association.

7.5.1.2 Role Ownership

Each instance of CIM_Role implemented as defined in sections 7.5.2.1, 7.5.3.1, and 7.5.4.1 shall be associated with the Scoping Instance through the CIM_OwningCollectionElement association.

7.5.2 Representing the CLP Administrator Role and Privileges

This section details the requirements for representing an SM CLP Administrator Role.

7.5.2.1 Administrator Role

There shall be an instance of CIM_Role implemented as specified in section 0.

7.5.2.2 Administrator Privileges

There shall be an instance of CIM_Privilege associated with the instance of CIM_Role defined in section 7.5.2.1 through the CIM_MemberOfCollection association. There shall not be any other instances of CIM_Privilege associated with instance of CIM_Role defined in section 7.5.2.1. The instance of CIM_Privilege shall be implemented as specified in section 10.21.

7.5.2.2.1 CIM_Privilege.Activities

For each array index of the CIM_Privilege.ActivityQualifiers property that contains a value, the corresponding index of the CIM_Privilege.Activities property shall contain the value 7 (Execute).

7.5.2.2.2 CIM_Privilege.ActivityQualifiers

The CIM_Privilege.ActivityQualifiers property shall contain the values {“cd”, “exit”, “help”, “show, “version”, “reset”, “start”, “stop”, “set”, “load”, “dump”, “create”, “delete”} and may contain additional values.

7.5.3 Representing the CLP Operator Role and Privileges (Optional)

This section details the requirements for representing an SM CLP User Role.

7.5.3.1 Operator Role

There shall be an instance of CIM_Role implemented as specified in section 10.28.

7.5.3.2 Operator Privileges

There shall be an instance of CIM_Privilege associated with the instance of CIM_Role defined in section 7.5.3.1 through the CIM_MemberOfCollection association. There shall not be any other instances of CIM_Privilege associated with instance of CIM_Role defined in section 7.5.3.1. The instance of CIM_Privilege shall be implemented as specified in section 10.22.

7.5.3.2.1 CIM_Privilege.Activities

For each array index of the CIM_Privilege.ActivityQualifiers property that contains a value, the corresponding index of the CIM_Privilege.Activities property shall contain the value 7 (Execute).

7.5.3.2.2 CIM_Privilege.ActivityQualifiers

The CIM_Privilege.ActivityQualifiers property shall contain the values {“cd”, “exit”, “help”, “show, “version”, “reset”, “start”, “stop”, “set”, “load”, “dump”}. The CIM_Privilege.ActivityQualifiers property shall not contain the values {“create”, “delete”}. The CIM_Privilege.ActivityQualifiers property may contain additional values.

7.5.4 Representing the CLP Read Only Role and Privileges

This section details the requirements for representing an SM CLP Read Only Role.

7.5.4.1 Read Only Role

There shall be an instance of CIM_Role implemented as specified in section 10.29.

7.5.4.2 Read Only Privileges

There shall be an instance of CIM_Privilege associated with the instance of CIM_Role defined in section 7.5.4.1 through the CIM_MemberOfCollection association. There shall not be any other instances of CIM_Privilege associated with instance of CIM_Role defined in section 7.5.4.1. The instance of CIM_Privilege shall be implemented as specified in section 10.23.

7.5.4.2.1 CIM_Privilege.Activities

For each array index of the CIM_Privilege.ActivityQualifiers property that contains a value, the corresponding index of the CIM_Privilege.Activities property shall contain the value 7 (Execute).

7.5.4.2.2 CIM_Privilege.ActivityQualifiers

The CIM_Privilege.ActivityQualifiers property shall contain the values {“cd”, “exit”, “help”, “show, “version”}. The CIM_Privilege.ActivityQualifiers property shall not contain the values {“reset”, “start”, “stop”, “set”, “load”, “dump”, “create”, “delete”}. The CIM_Privilege.ActivityQualifiers property may contain additional values.

7.5.5 CLP Security Principals

An instance of CIM_Identity representing the security principal of the CLP user shall exist or be instantiated when a CLP session is established and a CLP user has been authorized.

7.5.5.1 Identity Context

An instance of CIM_Identity that represents the security principal of a CLP User shall be associated with the Central Instance of this profile through the CIM_IdentityContext association.

7.5.5.2 Security Principal for Session — Optional

When the optional behavior specified in section 7.3 is implemented, the instance of CIM_Identity that corresponds to the security principal that was authenticated when the transport session underlying the CLP session was established shall be associated with the instance of CIM_TelnetProtocolEndpoint or CIM_SSHProtocolEndpoint that represents the transport session through the CIM_ConcreteDependency association. The value of the CIM_ConcreteDependency.Antecedent property shall be a reference to the instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint.

7.5.6 CLP Authorized Role Management

There shall be an instance of CIM_RoleBasedAuthorizationService associated to each instance of CIM_Role implemented as defined in sections 7.5.2, 7.5.3, and 7.5.4 through the CIM_ServiceAffectsElement association.

The instance of CIM_RoleBasedManagementCapabilities associated with the instance of CIM_RoleBasedAuthorizationService shall follow these requirements:

- The CIM_RoleBasedManagementCapabilities.SharedPrivilegeSupported property shall be set FALSE.
- The CIM_RoleBasedManagementCapabilities.ActivitiesSupported property shall have value of {7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute), 7(Execute)}.
- The CIM_RoleBasedManagementCapabilities.ActivityQualifiersSupported property shall have value {“cd”, “exit”, “help”, “show, “version”, “reset”, “start”, “stop”, “set”, “load”, “dump”, “create”, “delete”}.
- The CIM_RoleBasedManagementCapabilities.QualifierFormatsSupported property shall have value {9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction), 9 (Command Line Instruction)}.
- The SupportedMethods property shall be a zero length array.

7.5.6.1 OEM Extensions

The CIM_RoleBasedManagementCapabilities.ActivityQualifiersSupported property shall contain a value that corresponds to each OEM SM CLP verb supported. The corresponding array index of the CIM_RoleBasedManagementCapabilities.ActivitiesSupported property shall have value of {7(Execute) and the corresponding array index of the CIM_RoleBasedManagementCapabilities.QualifierFormatsSupported property shall have value {9 (Command Line Instruction).

7.6 CLP Operations and Queue

This section describes the requirements for modeling the CLP job queue and operations.

7.6.1 Job Queue

There shall be exactly one instance of CIM_JobQueue associated with the Scoping Instance through the CIM_HostedJobDestination. The Central Instance shall be associated with the CIM_JobQueue instance through the CIM_ServiceAffectsElement association.

7.6.2 Representing a CLP Operation

Each CLP command job shall be modeled with an instance of CIM_ConcreteJob. The instance of CIM_ConcreteJob shall be associated with the instance of CIM_JobQueue through the CIM_JobDestinationJobs association.

When an instance of CIM_ConcreteJob is created to represent a CLP operation, the initial value of the CIM_ConcreteJob.TimeBeforeRemoval property shall be the datetime representation of the value of the CIM_CLPProtocolEndpoint.KeepTime property of the instance of CIM_CLPProtocolEndpoint that represents the session through which the CLP command that resulted in the CLP operation was received.

The killing of a job may be supported; this is job and implementation specific. See section 8.2.

7.6.2.1 CIM_ConcreteJob.JobState

When a CIM_ConcreteJob is created the JobState property shall have the value 4 (Running). When a CLP operation completes successfully, the JobState property shall have the value 7 (Completed). When a CLP operation fails, the JobState property shall have the value 10 (Exception). When a CLP Operation is killed using the CIM_ConcreteJob.RequestStateChange() method where the RequestedState parameter has the value 5 (Kill), the JobState property shall have the value 9 (Killed). When a CLP Operation is terminated using the CIM_ConcreteJob.RequestStateChange() method where the RequestedState parameter has the value 5 (Terminate), the JobState property shall have the value 8 (Terminated).

7.6.2.2 CIM_ConcreteJob.ElementName

The ElementName property of an instance of CIM_ConcreteJob shall contain the Job Id for the modeled CLP operation. The property shall contain one or more digits matching the pattern "(1234567890)+".

7.6.3 Representing a CLP Operation Error

When a CLP operation fails, the CIM_ConcreteJob.JobState property of the CIM_ConcreteJob that represents the operation shall have the value 10 (Exception). The CIM_ConcreteJob.OperationalStatus property shall have a value other than 2 (Ok). The CIM_ConcreteJob.GetError() method can be used to retrieve an instance of CIM_Error detailing the cause of the operation failure. See section 8.3 for information on CIM_ConcreteJob.GetError().

7.6.3.1 Representing a Message (Optional)

An instance of CIM_Error may convey a Standard Message or vendor defined message. When an instance of CIM_Error is used to convey a message, the OwningEntity, MessageID, and Message properties shall be implemented and the MessageArguments property may be implemented. When an instance of CIM_Error is not used to convey a message, the OwningEntity, MessageID, Message, and MessageArguments properties shall not be implemented.

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_ProtocolService.RequestStateChange( )

CIM_ProtocolService.RequestStateChange( ) method invocation will change the element’s state to the value specified in the RequestedState parameter. The Enabled and Disabled values of the RequestedState parameter correspond to enabling or disabling the functionality represented by the instance of CIM_ProtocolService. A value of 2 (Enabled) shall correspond to a request to enable the functionality. A value of 3 (Disabled) shall correspond to a request to disable the functionality. A value of 11 (Reset) shall initiate a reset of the CLP service.

See section 7.1.3 for information about the effect of this method on the RequestedState property.

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

See section 7.1.3.4 for information about the effect of this method on the EnabledState property.

Detailed requirements of the RequestStateChange( ) method are specified in Table 2 and Table 3.

No standard messages are defined.

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

Table 2 – CIM_ProtocolService.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.

Table 3 – CIM_ProtocolService.RequestStateChange( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN,	RequestedState	uint16	Valid state values : 2 (Enabled) 3 (Disabled) 11 (Reset)
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

8.1.1 CIM_ProtocolService.RequestStateChange( ) ConditionalSupport

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

8.2 CIM_ConcreteJob.RequestStateChange()

The CIM_ConcreteJob.RequestStateChange( ) may be used to request that the CLP operation modeled by the CIM_ConcreteJob instance is terminated. See section 7.6.2.1 for the effect of this method on the JobState property.

No standard messages are defined.

Table 4 – CIM_ConcreteJob.RequestStateChange( ) Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred.

Table 5 – CIM_ConcreteJob.RequestStateChange( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	RequestedState	uint16	Valid state values : 5 (Kill) 4 (Terminate)

Note that the use of the TimeoutPeriod parameter is not supported.

8.3 CIM_ConcreteJob.GetError()

The GetError() method is used to retrieve the instance of CIM_Error that contains details of why the operation failed if it did. This method shall be supported when the CIM_ConcreteJob.JobStatus has the value 10 (Exception) and shall not return a value of 1.

No standard messages are defined.

Table 6 – CIM_ConcreteJob.GetError( ) Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported in the implementation.
2	Error occurred.

Table 7 – CIM_ConcreteJob.GetError( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
OUT	Error	String	Embedded instance of CIM_Errro

8.4 Profile Conventions for Operations

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

The default list of operations is as follows:

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

8.5 CIM_BindsTo

Table 8 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 8, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 8 – Operations: CIM_BindsTo

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

8.6 CIM_CLPCapabilities

All operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

8.7 CIM_CLPSettingData

Table 9 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 9, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 9 – Operations: CIM_CLPSettingData

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.7.1.	None

8.7.1 CIM_CLPSettingData – ModifyInstance

When the CIM_CLPSettingData instance is associated with the CIM_ProtocolService instance through an instance of CIM_ElementSettingData and the value of the IsDefault property of the CIM_ElementSettingData instance that associates the CIM_CLPSettingData with the CIM_ProtocolService has a value of 1 (Is Default), the ModifyInstance operation shall not be supported.

When the CIM_CLPSettingData instance is not associated with an instance of CIM_ProtocolService through an instance of CIM_ElementSettingData where the IsDefault property has a value of 1 (Is Default), the ModifyInstance operation may be supported for the CIM_CLPSettingData instance.

8.8 CIM_CLPProtocolEndpoint

Table 10 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 10, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 10 – Operations: CIM_CLPProtocolEndpoint

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.8.1.	None
DeleteInstance	Optional. See section 8.8.2.	None

8.8.1 ModifyInstance

The ModifyInstance operation may be supported for an instance of CIM_CLPProtocolEndpoint. When the ModifyInstance operation is supported for an CIM_CLPProtocolEndpoint instance, the ModifyInstance operation shall not modify the following properties:

- NameFormat
- ProtocolIFType
- OtherTypeDescription

8.8.2 DeleteInstance

The DeleteInstance operation may be supported for instances of CIM_CLPProtocolEndpoint. When the DeleteInstance operation is invoked against an instance, the corresponding CLP session shall be terminated prior to deleting the CIM_CLPProtocolEndpoint instance. The implementation shall also remove any association instances that reference the CIM_CLPProtocolEndpoint.

8.9 CIM_ConcreteJob

Table 11 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 11, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 11 – Operations: CIM_ConcreteJob

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.9.1.	None
DeleteInstance	Optional. See section 8.9.2.	None

8.9.1 ModifyInstance

The ModifyInstance operation may be supported for an instance of CIM_ConcreteJob. When the ModifyInstance operation is supported, the TimeBeforeRemoval and DeleteOnCompletion properties shall be writable.

8.9.2 DeleteInstance

The DeleteInstance operation may be supported for an instance of CIM_ConcreteJob. When the DeleteInstance operation is supported, the DeleteInstance operation shall fail when the CIM_ConcreteJob.JobStatus property has the value 4 (Running).

8.10 CIM_ElementCapabilities

Table 12 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 12, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 12 – Operations: CIM_ElementCapabilities

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

8.11 CIM_ElementSettingData

Table 13 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 13, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 13 – Operations: CIM_ElementSettingData

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.11.1.	None
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.11.1 CIM_ElementSettingData – ModifyInstance

When an instance of CIM_ElementSettingData associates an instance of CIM_CLPSettingData with an instance of CIM_CLPProtocolEndpoint, the following rules shall govern the behavior of the ModifyInstance operation:

- The ModifyInstance operation shall not allow the IsDefault property to be modified.
- The ModifyInstance operation shall not allow the IsCurrent property to be modified.
- When the ModifyInstance operation is used to modify the IsNext property to have a value of 1 (Is Next), the ModifyInstance operation shall implement the following behavior:
  - The ModifyInstance operation shall find all other instances of CIM_ElementSettingData that associate an CIM_CLPSettingData instance with the CIM_CLPProtocolEndpoint instance referenced by the target instance of CIM_ElementSettingData.
  - For each 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).

8.12 CIM_Error

The sole usage of CIM_Error in this profile is as a template for an embedded instance. Therefore, none of the operations in the default list in 8.4 shall be supported as described by DSP0200.

8.13 CIM_HostedAccessPoint

Table 14 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 14, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 14 – Operations: CIM_HostedAccessPoint

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

8.14 CIM_HostedJobDestination

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 8.4 shall be implemented as defined in DSP0200.

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

Table 15 – Operations: CIM_HostedJobDestination

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

8.15 CIM_HostedService

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 8.4 shall be implemented as defined in DSP0200.

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

Table 16 – Operations: CIM_HostedService

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

8.16 CIM_JobDestinationJobs

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 8.4 shall be implemented as defined in DSP0200.

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

Table 17 – Operations: CIM_JobDestinationJobs

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

8.17 CIM_JobQueue

All operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

8.18 CIM_OwningJobElement

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 8.4 shall be implemented as defined in DSP0200.

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

Table 18 – Operations: CIM_OwningJobElement

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

8.19 CIM_ProtocolService

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 8.4 shall be implemented as defined in DSP0200.

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

Table 19 – Operations: CIM_ProtocolService

Operation	Requirement	Messages
ModifyInstance	Optional. See section 8.19.1.	None

8.19.1 CIM_ProtocolService — ModifyInstance

When the ElementNameEditSupported property of the CIM_CLPCapabilities has a value of TRUE, the ModifyInstance operation shall allow the value of the ElementName property of the CIM_ProtocolService instance to be modified. The ModifyInstance operation shall enforce the length restriction specified in the MaxElementNameLen property of the CIM_CLPCapabilities.

When the ElementNameEditSupported property of the CIM_CLPCapabilities has a value of FALSE, the ModifyInstance operation shall not change the value of the ElementName property of the CIM_ProtocolService instance.

8.20 CIM_ProvidesEndpoint

Table 20 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 20, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 20 – Operations: CIM_ProvidesEndpoint

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

8.21 CIM_ServiceAccessBySAP

Table 21 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 21, all operations in the default list in 8.4 shall be implemented as defined in DSP0200.

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

Table 21 – Operations: CIM_ServiceAccessBySAP

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

8.22 ServiceAffectsElement

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 8.4 shall be implemented as defined in DSP0200.

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

Table 22 – Operations: CIM_ServiceAffectsElement

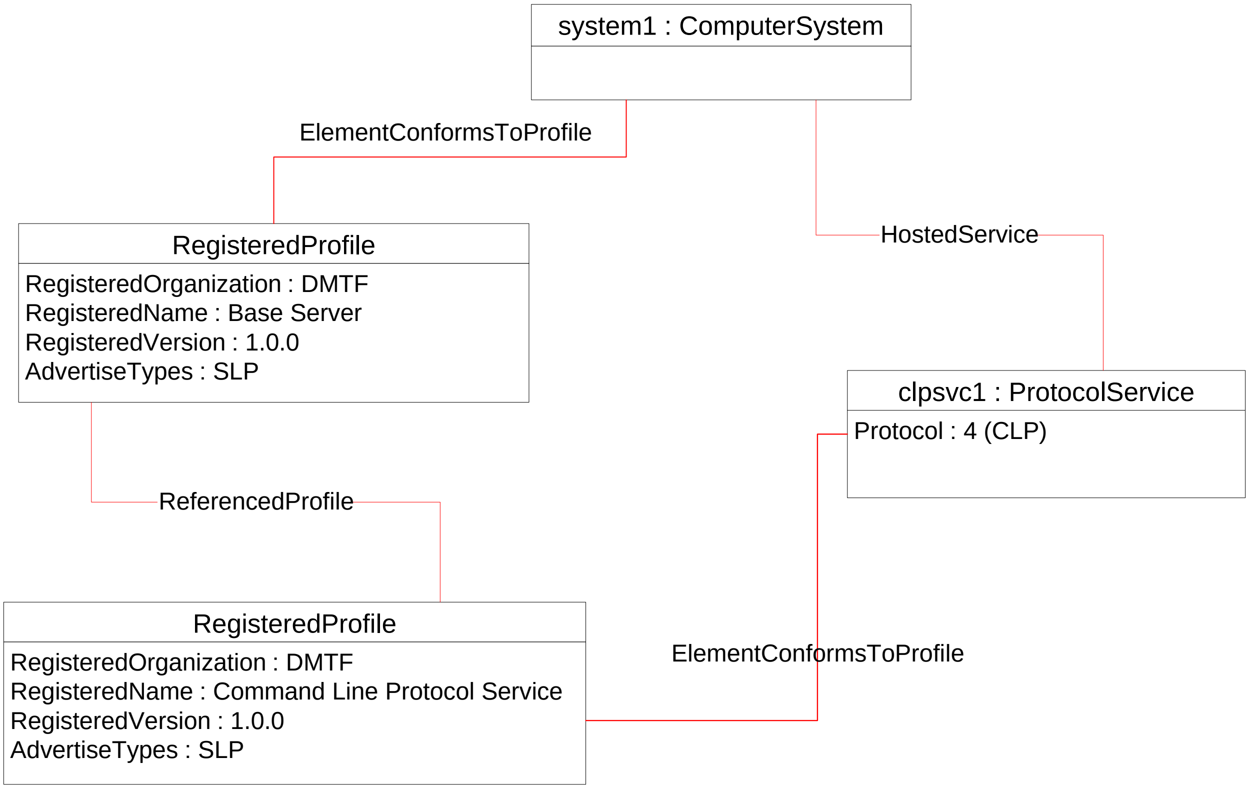
Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

9 Use Cases

The following sections outline common use cases for client interaction with the *Command Line Protocol Service Profile*.

9.1 Object Diagrams

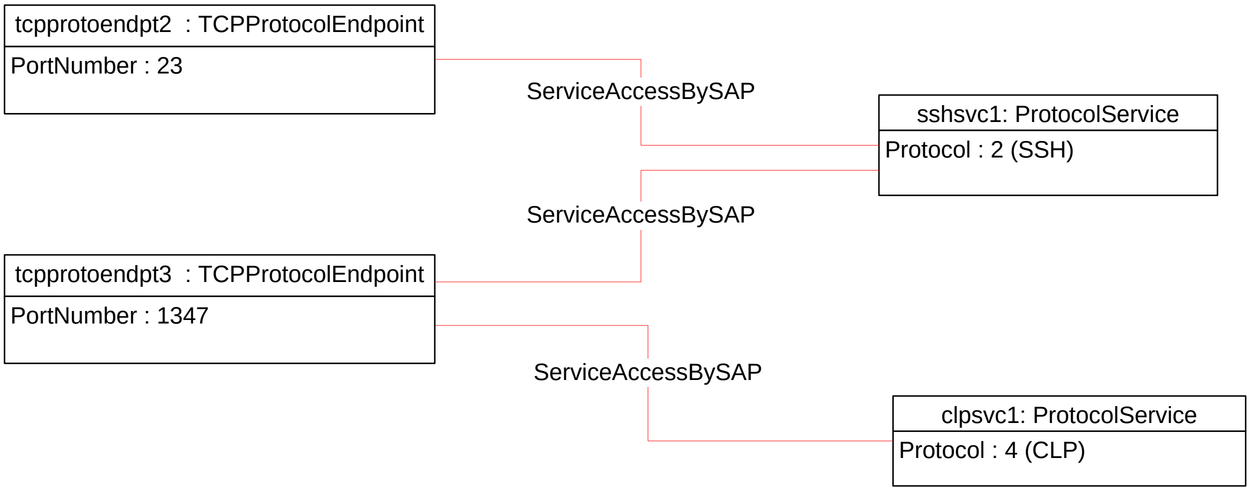
The object diagram in Figure 2 shows how instances of CIM_RegisteredProfile are used to identify the version of the *Command Line Protocol Service Profile* with which an instance of CIM_ProtocolService and its associated instances are conformant. An instance of CIM_RegisteredProfile exists for each profile that is instrumented in the system. One instance of CIM_RegisteredProfile identifies the “DMTF *Base Server Profile* version 1.0”. The other instance identifies the “DMTF *Command Line Protocol Service Profile* version 1.0”. The CIM_ProtocolService instance is scoped to an instance of CIM_ComputerSystem. This instance of CIM_ComputerSystem is conformant with the DMTF *Base Server Profile* version 1.0 as indicated by the CIM_ElementConformsToProfile association to the CIM_RegisteredProfile instance. This implementation is using the Central Class Methodology. The CIM_ProtocolService instance is conformant with this profile as indicated by the CIM_ElementConformsToProfile association between the instance and the instance of CIM_RegisteredProfile that identifies this profile.


system1 : ComputerSystem
ElementConformsToProfile
HostedService
RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Base Server
RegisteredVersion : 1.0.0
AdvertiseTypes : SLP
clpsvc1 : ProtocolService
Protocol : 4 (CLP)
ReferencedProfile
RegisteredProfile
RegisteredOrganization : DMTF
RegisteredName : Command Line Protocol Service
RegisteredVersion : 1.0.0
AdvertiseTypes : SLP
ElementConformsToProfile

861

862 **Figure 2 – Registered Profile**

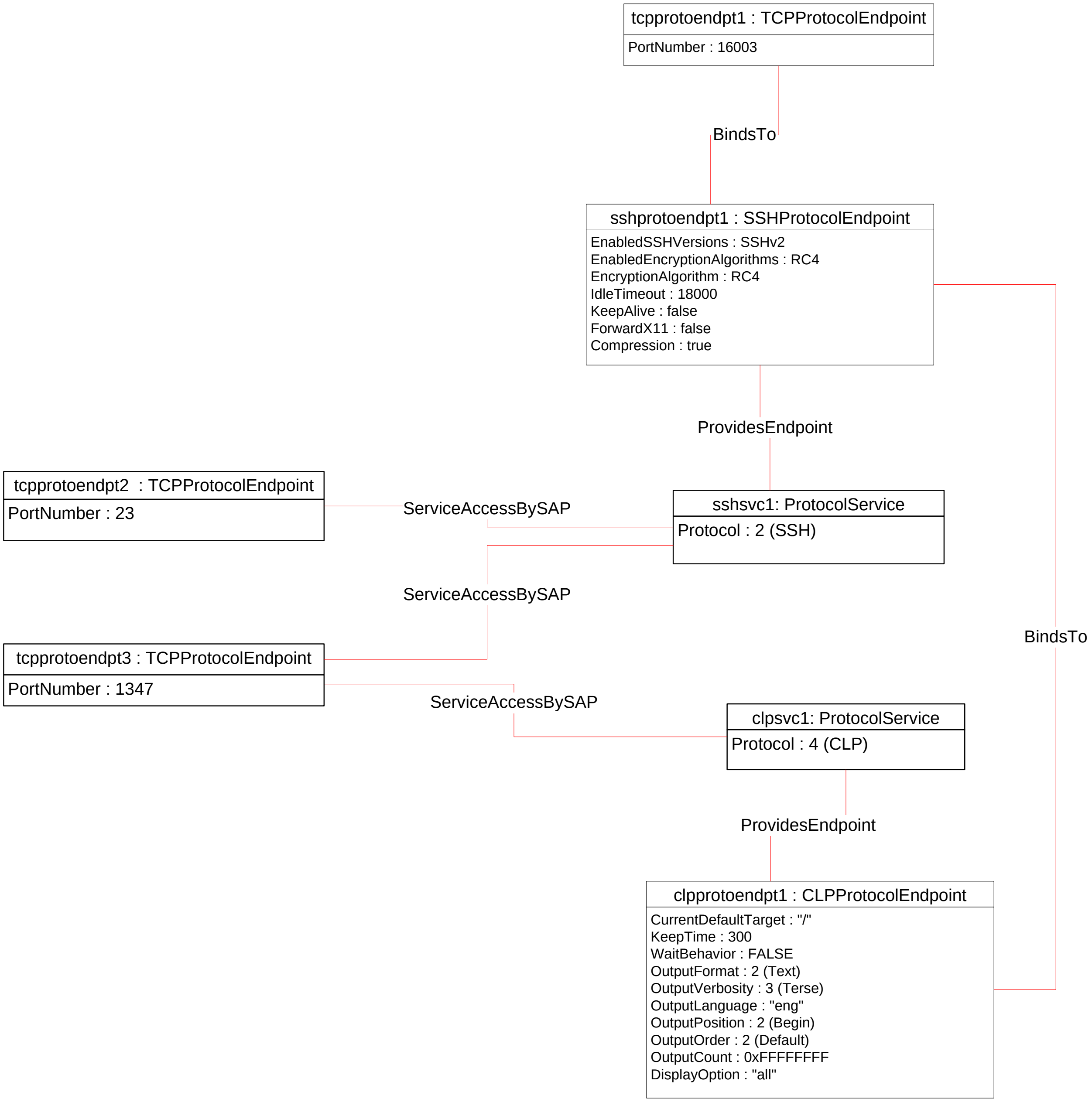
863 Figure 3 is an object diagram illustrating a CLP service accessible through SSH. The SSH service is
864 listening on ports 23 and 1347. The CLP service is accessible through SSH sessions established by
865 connecting to port 1347. This is indicated by the CIM_ServiceAccessBySAP between the `clpsv1` and
866 `tcpprotoendpt3`. The object diagram does not indicate whether the CLP session is automatically
867 initiated upon establishment of the SSH session or requires a manual step on the part of the user.



868

869 **Figure 3 – CLP Service Accessible via SSH**

870 Figure 4 illustrates a single CLP session (`clpprotoendpt1`) established through an SSH session
871 (`sshprotoendpt1`). This is indicated by the CIM_BindsTo association that references the two instances.



872

873 **Figure 4 – One Active Session**

The object diagram in Figure 5 provides a more complete view of a CLP implementation that is capable of managing two systems. The manageable systems are each represented by an instance of CIM_ComputerSystem aggregated into the CIM_AdminDomain instance. The CIM_ServiceAffectsElement association identifies the CIM_AdminDomain as the management domain of this CLP service. A single CLP session is active and there are no CLP operations currently executing.

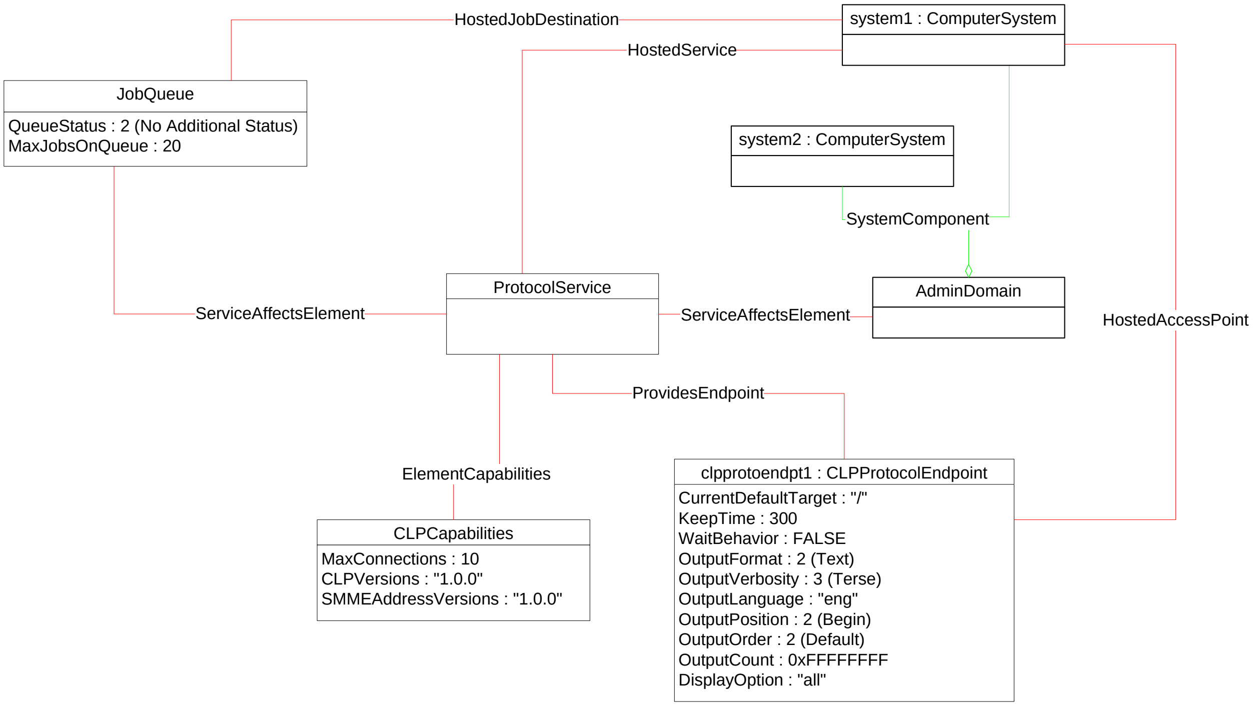


Figure 5 – Session with Admin Domain and Job Queue

Figure 6 is an object diagram for the same CLP service as in Figure 5. Note that the KeepTime property on `clpprotoendpt1` has been changed from the specification default value of 300 seconds to a new value of 600 seconds prior to the initiation of the CLP operation which is modeled by `job454`. The CLP operation represented by `job454` is currently executing as indicated by the value of the JobState property.

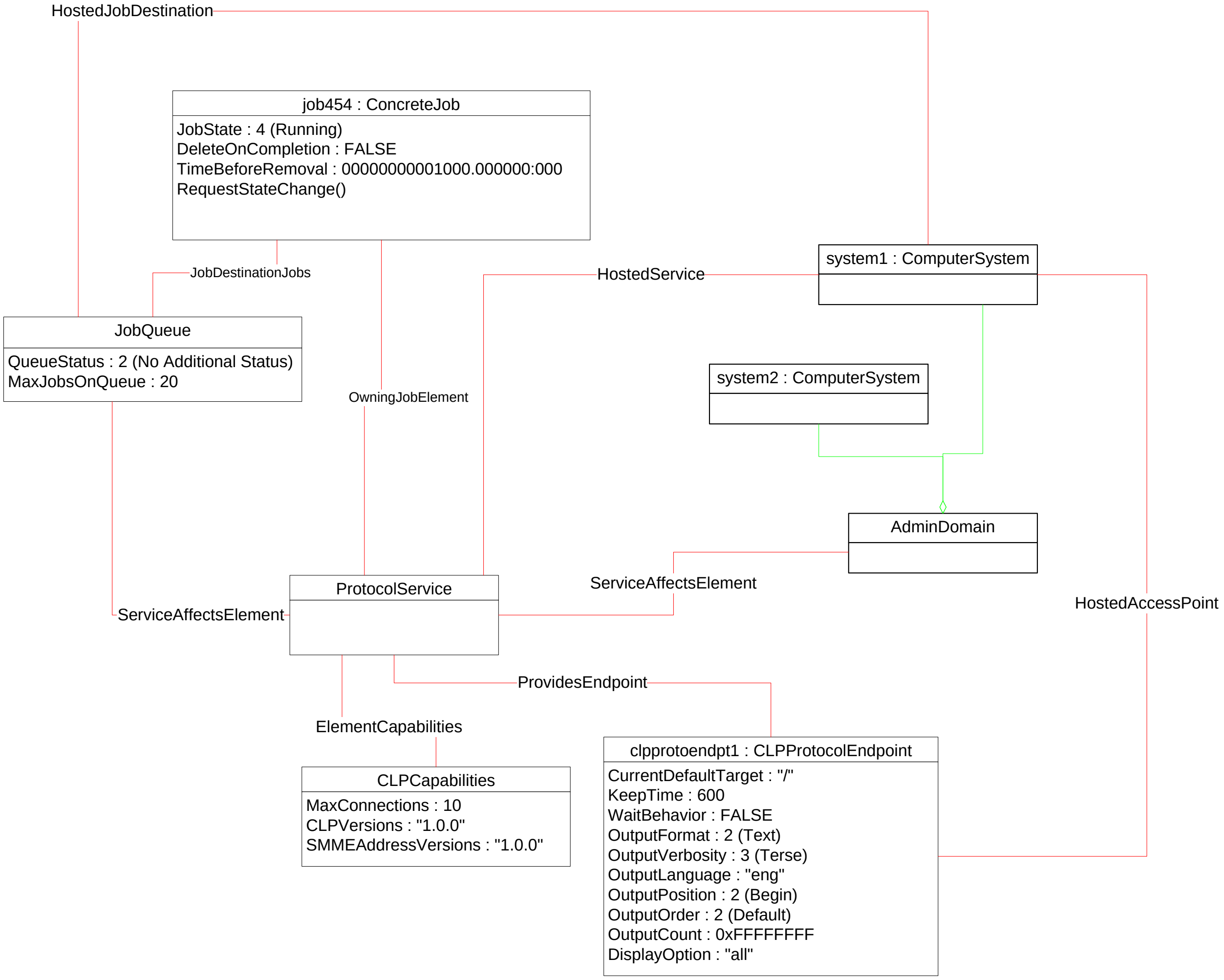
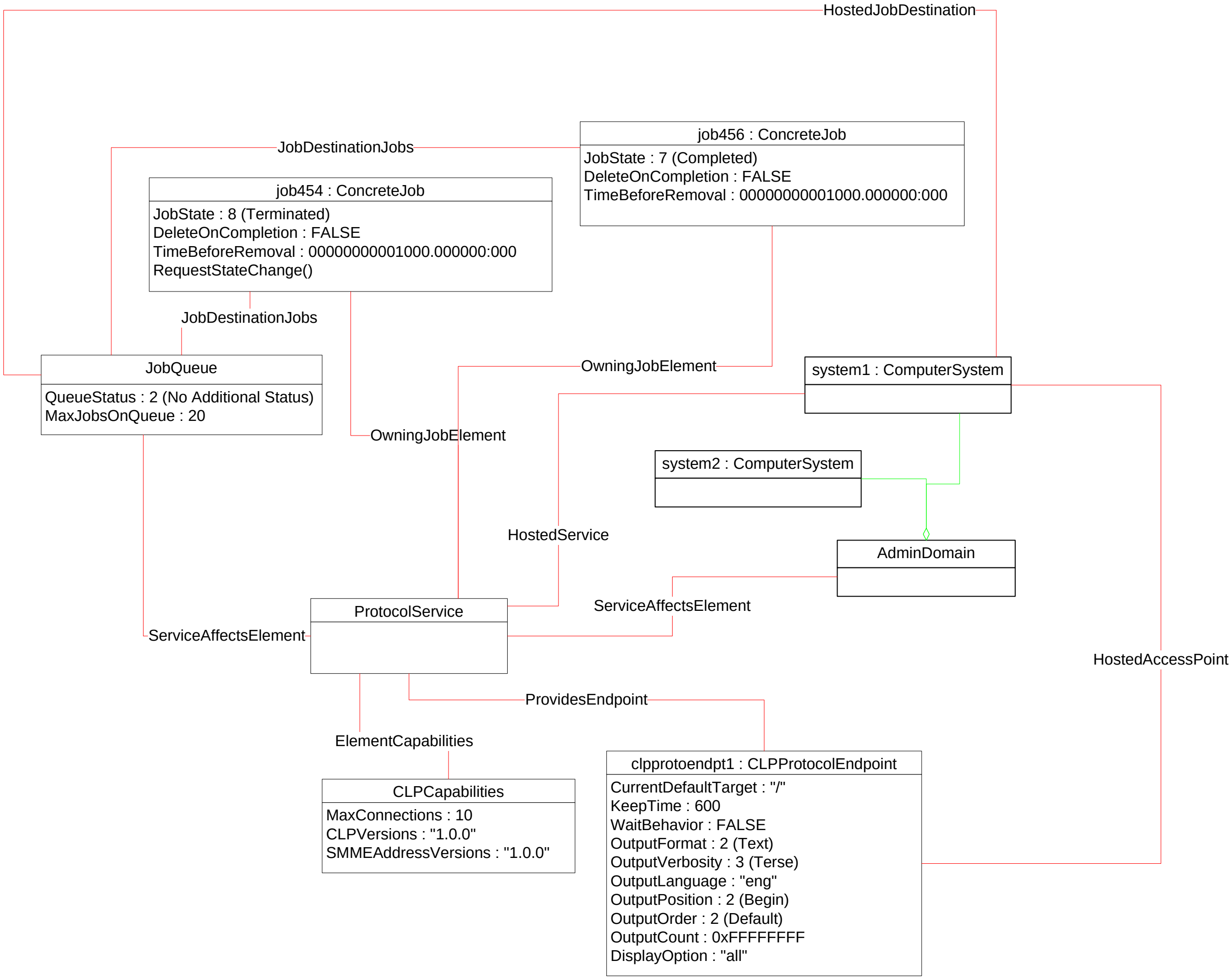


Figure 6 – Single Operation Executing

888 Figure 7 is an object diagram for the same CLP service as in Figure 6. Prior to the completion of the CLP
889 operation represented by `job454` another CLP command was issued to terminate the first operation. The
890 latter CLP command resulted in the CLP operation modeled by `job456`. This operation successfully
891 completed as indicated by the value of 7 (Completed) for its JobState property. The fact that the
892 previously initiated CLP operation is reflected by a value of 8 (Terminated) for the JobState property of
893 `job454`.



894

895 **Figure 7 – CLP Command Used to Terminate Running CLP Operation**

Figure 8 is an object diagram showing a subset of an implementation related to management of CLP roles. There are two instances of CIM_Role representing the Administrator and Read Only roles. There are two accounts on the system. Each account has a corresponding security principal represented by CIM_Identity whose context includes usage by the CLP service.

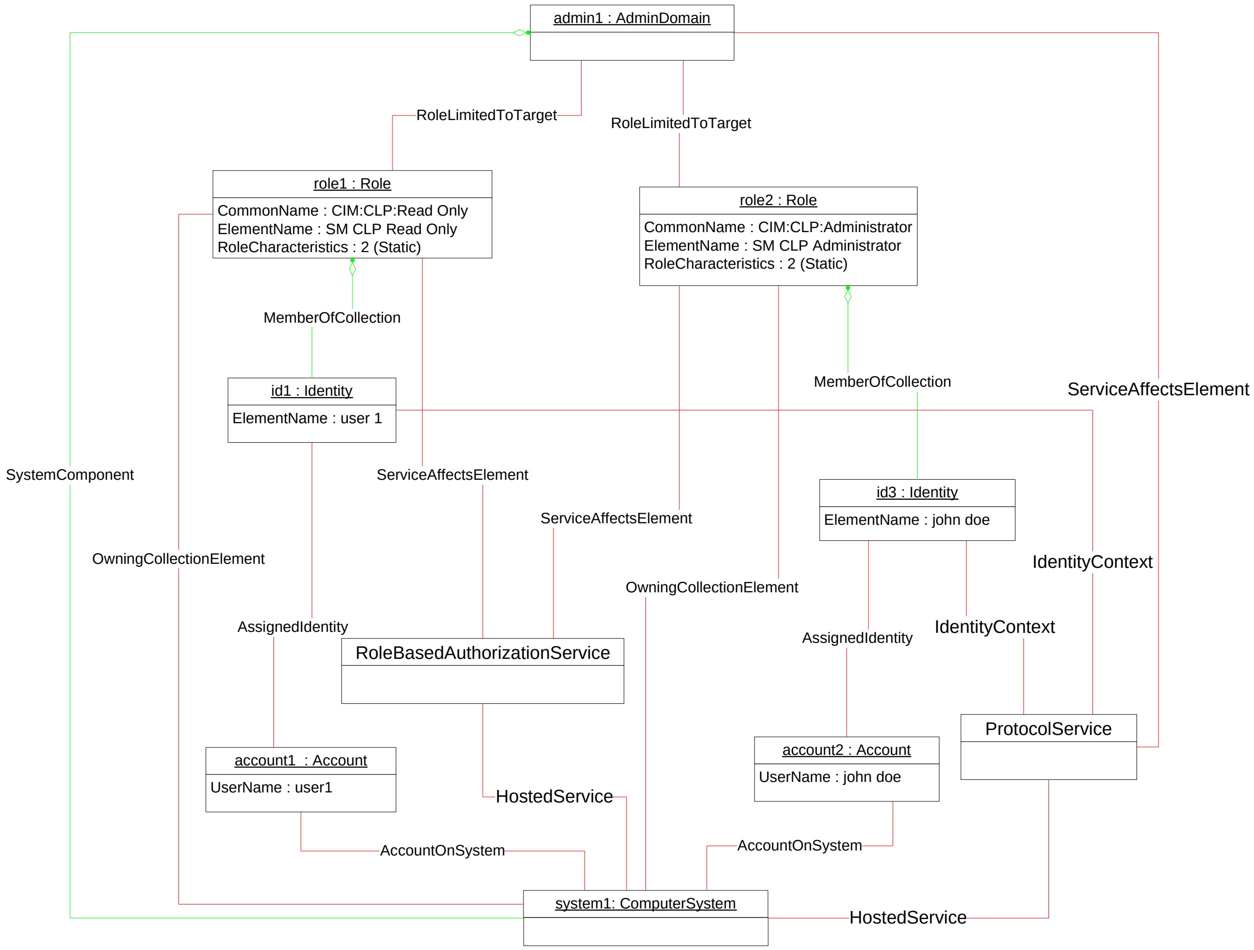


Figure 8 – CLP Roles

Figure 9 is an object diagram showing an active CLP session that has been established over SSH. `id3` is the corresponding security principal that resulted from the credentials provided when the SSH session was established. This is indicated by the CIM_ConcreteDependency association between the `sshprotoendpt1` and `id3`. The credentials provided correspond to `account2`. This is indicated by the CIM_AssignedIdentity instance that associated `id3` with `account2`. The user of the CLP session has Administrator rights. This is determined by association traversal from `clpprotoendpt1` to `sshprotoendpt1` through the CIM_BindsTo association, then to `id3` across the CIM_ConcreteDependency association, then to `role2` through the CIM_MemberOfCollection association. The account for `id1` is not shown.

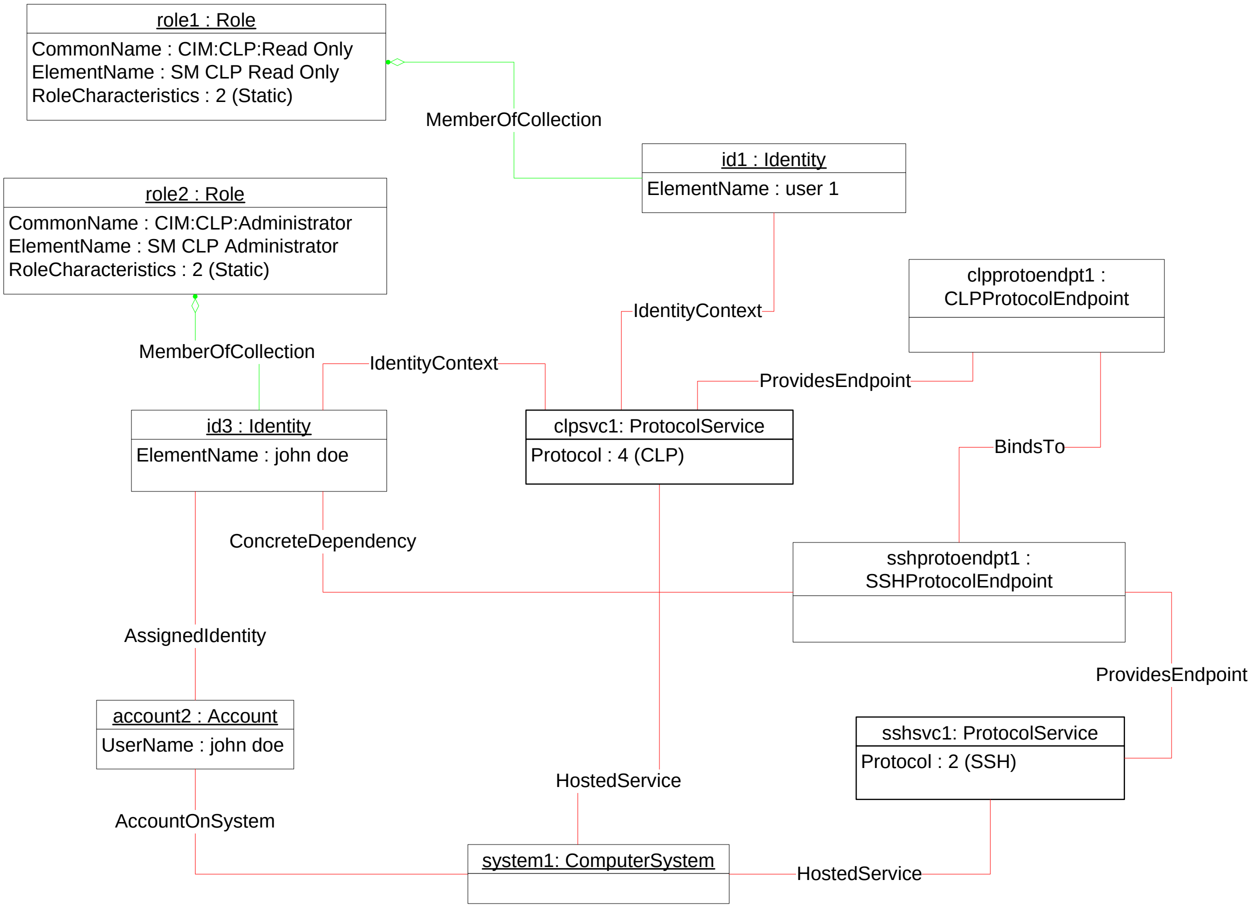


Figure 9 – Active Session with Security Principal

9.2 Modifying Active Session Settings

A user can find the active sessions for a CLP service and modify their configuration as follows:

1) Find an instance of CIM_CLPProtocolEndpoint associated with the CIM_ProtocolService through an instance of CIM_ProvidesEndpoint.

2) Modify the properties of the CIM_CLPProtocolEndpoint as desired.

9.3 Disabling the CLP Service

If an implementation supports disabling the CLP service, a user can disable the CLP service by invoking the RequestStateChange( ) method on CIM_ProtocolService instance with a value of Disabled for the RequestedState parameter.

9.4 Determining the CLP Service Capabilities

A user can determine the capabilities of the CLP service as follows:

1) Find the instance of CIM_CLPCapabilities associated with the CIM_ProtocolService through an instance of CIM_ElementCapabilities.

2) View the properties of the CIM_CLPCapabilities instance to see the supported function.

9.5 Determining If ElementName Can Be Modified

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

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

2) Query the value of the ElementNameEditSupported property of the CIM_CLPCapabilities instance. If the value is TRUE, the client can modify the ElementName property of the target instance.

9.6 Determining If State Management Is Supported

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

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

2) Query the value of the RequestedStatesSupported property. If at least one value is specified, state management is supported.

10 CIM Elements

Table 23 shows the instances of CIM Elements for this profile. Instances of these CIM Elements shall be implemented as described in Table 23. Section 7 may impose additional requirements on these elements.

Table 23 – CIM Elements: Command Line Protocol Service Profile

Element Name	Requirement	Notes
Classes		
CIM_BindsTo	Optional	See section 10.1.
CIM_CLPCapabilities	Mandatory	See section 10.2.
CIM_CLPProtocolEndpoint	Mandatory	See section 10.3
CIM_CLPSettingData	Optional	See sections 10.4 and 10.5.
CIM_ConcreteDependency	Optional	See section 10.6.
CIM_ConcreteJob	Mandatory	See section 10.7.
CIM_ElementCapabilities	Mandatory	See section 10.8.
CIM_ElementSettingData	Optional	See sections 10.9 and 10.10.
CIM_Error	Mandatory	See section 10.11.

Element Name	Requirement	Notes
CIM_HostedAccessPoint	Mandatory	See section 10.12.
CIM_HostedJobDestination	Mandatory	See section 10.13.
CIM_HostedService	Mandatory	See section 10.14.
CIM_JobDestinationJobs	Conditional	See section 10.15.
CIM_JobQueue	Mandatory	See section 10.17.
CIM_OwningCollectionElement	Mandatory	See section 10.18.
CIM_OwningJobElement	Conditional	See sections 10.19 and 10.20.
CIM_Privilege	Mandatory	See sections 10.21, 10.22, and 10.23.
CIM_ProtocolService	Mandatory	See section 10.24.
CIM_ProvidesEndpoint	Mandatory	See section 10.25.
CIM_RegisteredProfile	Mandatory	See section 10.26.
CIM_Role	Mandatory	See sections 10.27, 10.28, and 10.29.
CIM_RoleLimitedToTarget	Mandatory	See section 10.30.
CIM_ServiceAccessBySAP	Optional	See section 10.31.
CIM_ServiceAffectsElement	Mandatory	See sections 10.32 and 10.33.
Indications		
None defined in this profile		

10.1 CIM_BindsTo

CIM_BindsTo is used to relate the CIM_CLPProtocolEndpoint instance with the CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint instance on which it is dependent.

Table 24 – Class: CIM_BindsTo

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** shall be a reference to an instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint. Cardinality 0..1
Dependent	Mandatory	**Key** The value of this property shall be a reference to an instance of CIM_CLPProtocolEndpoint. Cardinality 1

10.2 CIM_CLPCapabilities

CIM_CLPCapabilities represents the capabilities of a CLP service.

Table 25 – Class: CIM_CLPCapabilities

Properties	Requirement	Notes
InstanceID	Mandatory	**Key**
ElementName	Mandatory	(pattern ".*")
RequestedStatesSupported	Mandatory	See section 7.2.1.
ElementNameEditSupported	Mandatory	See section 7.4.1.
MaxElementNameLen	Conditional	See section 7.1.4.2.2.
MaxConnections	Mandatory	See section 7.1.2.1.
CLPVersions	Mandatory	Shall contain one or more occurrences pattern ([0123456789] "\." [0123456789] "\." [0123456789]?)
SMMEAddressVersions	Mandatory	Shall contain one or more occurrences pattern ([0123456789] "\." [0123456789] "\." [0123456789]?)

10.3 CIM_CLPProtocolEndpoint

CIM_CLPProtocolEndpoint represents a session established with the CLP service. There shall be an instance of CLPProtocolEndpoint for each remote access point provided by the CLP Service to a Client.

Table 26 – Class: CIM_CLPProtocolEndpoint

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
NameFormat	Mandatory	pattern ".*"
ProtocolIFType	Mandatory	Matches 1 (Other)
OtherTypeDescription	Mandatory	Matches "CLP"
ElementName	Mandatory	pattern ".*"
CurrentDefaultTarget	Mandatory	pattern ".+"
KeepTime	Mandatory	None
WaitBehavior	Mandatory	None
OutputFormat	Mandatory	None
OutputVerbosity	Mandatory	None
OutputLanguage	Mandatory	None
OutputPosition	Mandatory	None
OutputOrder	Mandatory	None
OutputCount	Mandatory	None
DisplayOption	Mandatory	pattern ".+"

10.4 CIM_CLPSettingData – Specification Default Configuration

CIM_CLPSettingData represents settings which can be applied to a CLP session.

Table 27 – Class: CIM_CLPSettingData

Properties	Requirement	Notes
InstanceId	Mandatory	**Key**
ElementName	Mandatory	Matches "Specification Defaults"
CommandDefaultTarget	Mandatory	Matches "/"
KeepTime	Mandatory	Matches 300
WaitBehavior	Mandatory	Matches FALSE
OutputFormat	Mandatory	Matches 2 (Text)
OutputVerbosity	Mandatory	Matches 3 (Terse)
OutputLanguage	Mandatory	Matches "eng"
OutputPosition	Mandatory	Matches 2 (Begin)
OutputOrder	Mandatory	Matches 2 (Default)
OutputCount	Mandatory	Matches 0xFFFFFFFF
DisplayOption	Mandatory	Matches "all"

10.5 CIM_CLPSettingData

CIM_CLPSettingData represents settings which can be applied to a CLP session. There shall be one default CLPSettingData instance containing the default settings for the configurable properties of the CLP Service. There may be other CLPSettingData instances describing optional configurations.

Table 28 – Class: CIM_CLPSettingData

Properties	Requirement	Notes
InstanceId	Mandatory	**Key**
ElementName	Mandatory	pattern ".*"
CommandDefaultTarget	Mandatory	pattern ".+"
KeepTime	Mandatory	None
WaitBehavior	Mandatory	None
OutputFormat	Mandatory	None
OutputVerbosity	Mandatory	None
OutputLanguage	Mandatory	None
OutputPosition	Mandatory	None
OutputOrder	Mandatory	None
OutputCount	Mandatory	None
DisplayOption	Mandatory	pattern ".+"

10.6 CIM_ConcreteDependency – (Access Ingress)

Table 29 details the constraints for instances of CIM_ConcreteDependency beyond those specified in the DSP1034.

Table 29 – Class: CIM_ConcreteDependency – (Access Ingress)

Elements	Requirement	Notes
Antecedent	Mandatory	See section 7.5.5.2.

10.7 CIM_ConcreteJob

CIM_ConcreteJob represents a single CLP command job.

Table 30 – Class: CIM_ConcreteJob

Properties	Requirement	Notes
ElementName	Mandatory	See section 7.6.2.2.
InstanceId	Mandatory	**Key**
JobState	Mandatory	Matches 4 | 7 | 8 | 9 |10 (Running | Completed | Terminated | Killed | Exception) See section 7.6.2.1.
DeleteOnCompletion	Mandatory	matches TRUE
TimeBeforeRemoval	Mandatory	See section 7.6.2.
RequestStateChange()	Mandatory	See section 8.2.
GetError()	Conditional	See section 8.3.
OperationalStatus	Mandatory	See section 7.6.3.

10.8 CIM_ElementCapabilities

CIM_ElementCapabilities is used to associate an instance of CIM_CLPCapabilities with the CIM_ProtocolService

Table 31 – Class: CIM_ElementCapabilities

Properties	Requirement	Notes
ManagedElement	Mandatory	**Key** This property shall be a reference to the Central Instance. Cardinality 1..*
Capabilities	Mandatory	**Key** This property shall be a reference to the CIM_CLPCapabilities instance. Cardinality 1

10.9 CIM_ElementSettingData – CLP Service

CIM_ElementSettingData is used to associate instances of CIM_CLPSettingData with instances of CIM_ProtocolService.

Table 32 – Class: CIM_ElementSettingData (CLP Service)

Properties	Requirement	Notes
ManagedElement	Mandatory	**Key** This property shall be a reference to the Central Instance. Cardinality *
Setting	Mandatory	**Key** This property shall be a reference to an instance of CIM_CLPSettingData. Cardinality *
IsDefault	Mandatory	Matches 1 (Is Default) or 2 (Is Not Default)
IsNext	Mandatory	Matches 1 (Is Next) or 2 (Is Not Next)

10.10 CIM_ElementSettingData – CLP Session

CIM_ElementSettingData is used to associate instances of CIM_CLPSettingData with instances of CIM_CLPProtocolEndpoint.

Table 33 – Class: CIM_ElementSettingData (CLP Session)

Properties	Requirement	Notes
ManagedElement	Mandatory	**Key** This property shall be a reference to an instance of CIM_CLPProtocolEndpoint. Cardinality *
Setting	Mandatory	**Key** This property shall be a reference to an instance of CIM_CLPSettingData. Cardinality *
IsCurrent	Mandatory	Matches 1 (Is Current) or 2 (Is Not Current)

10.11 CIM_Error

CIM_Error is returned by the CIM_ConcreteJob.GetError() method when a CLP operation fails.

Table 34 – Class: CIM_Error

Properties	Requirement	Notes
ErrorType	Mandatory	None
OwningEntity	Conditional	See section 7.6.3.1.
MessageID	Conditional	See section 7.6.3.1.
Message	Conditional	See section 7.6.3.1.
MessageArguments	Conditional	See section 7.6.3.1.
PerceivedSeverity	Mandatory	None
ProbableCause	Mandatory	None
RecommendedActions	Optional	None

Properties	Requirement	Notes
ErrorSource	Mandatory	None
ErrorSourceFormat	Mandatory	None
CIMStatusCode	Mandatory	None

10.12 CIM_HostedAccessPoint

CIM_HostedAccessPoint is used to relate the CIM_CLPProtocolEndpoint and CIM_TCPProtocolEndpoint instances to their scoping CIM_ComputerSystem instance.

Table 35 – Class: CIM_HostedAccessPoint

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to an instance of CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	**Key** This property shall be a reference to an instance of CIM_CLPProtocolEndpoint or CIM_TCPProtocolEndpoint. Cardinality *

10.13 CIM_HostedJobDestination

CIM_HostedJobDestination is used to associate the single instance of CIM_JobQueue with hosting CIM_ComputerSystem instance.

Table 36 – Class: CIM_HostedJobDestination

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to the hosting CIM_ComputerSystem. Cardinality 1
Dependent	Mandatory	**Key** This property shall be a reference to the hosted CIM_JobQueue. Cardinality 1

10.14 CIM_HostedService

CIM_HostedService is used to relate the CIM_ProtocolService to its scoping CIM_ComputerSystem instance.

Table 37 – Class: CIM_HostedService

Properties	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..*

10.15 CIM_IdentityContext

Table 38 details the constraints for instances of CIM_IdentityContext beyond those specified in DSP1034.

Table 38 – Class: CIM_IdentityContext

Elements	Requirement	Notes
ElementProvidingContext	Mandatory	shall be a reference to the Central Instance Cardinality 1

10.16 CIM_JobDestinationJobs

CIM_JobDestinationJobs is used to associate instances of CIM_JobQueue with instances of CIM_ConcreteJob.

Table 39 – Class: CIM_JobDestinationJobs

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to the CIM_JobQueue. Cardinality 1
Dependent	Mandatory	**Key** This property shall be a reference to the CIM_ConcreteJob. Cardinality *

10.17 CIM_JobQueue

CIM_JobQueue represents the CLP Service's operations queue.

Table 40 – Class: CIM_JobQueue

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
Name	Mandatory	**Key**
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
ElementName	Mandatory	(pattern ".*")
MaxJobsOnQueue	Mandatory	A value of 0 (zero) shall indicate the maximum number of jobs is unknown or unenforced.
QueueStatus	Mandatory	None
QueueStatusInfo	Mandatory	None

10.18 CIM_OwningCollectionElement

Table 41 details the constraints for properties of CIM_OwningCollectionElement beyond those specified in the *Role Based Authorization Profile*.

Table 41 – Class: CIM_OwningCollectionElement

Properties and Methods	Requirement	Description
OwningElement	Mandatory	See section 7.5.1.2.

10.19 CIM_OwningJobElement – CLP Service

CIM_OwningJobElement is used to associate instances of CIM_ConcreteJob with the instance of CIM_ProtocolService representing the CLP Service.

Table 42 – Class: CIM_OwningJobElement

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to the hosting CIM_ProtocolService Cardinality 1
Dependent	Mandatory	**Key** This property shall be a reference to the hosted CIM_ConcreteJob. Cardinality *

10.20 CIM_OwningJobElement – CLP Session

CIM_OwningJobElement is used to associate instances of CIM_ConcreteJob with the instance of CIM_CLPProtocolEndpoint representing the CLP session where the CLP operation was initiated.

Table 43 – Class: CIM_OwningJobElement

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to the hosting CIM_CLPProtocolEndpoint Cardinality 1
Dependent	Mandatory	**Key** This property shall be a reference to the hosted CIM_ConcreteJob. Cardinality *

10.21 CIM_Privilege (Administrator)

An instance of CIM_Privilege is used to represent the privileges of the SM CLP Administrator Role. Table 44 details the constraints for properties of CIM_Privilege beyond those specified in the *Role Based Authorization Profile* when it used to model the privileges of the SM CLP Administrator Role.

Table 44 – Class: CIM_Privilege (Administrator)

Properties and Methods	Requirement	Description
PrivilegeGranted	Mandatory	Matches TRUE
Activities	Mandatory	See section 7.5.2.2.1.
ActivityQualifiers	Mandatory	See section 7.5.2.2.2.
ElementName	Mandatory	Matches "SM CLP Administrator Privilege"

10.22 CIM_Privilege (Operator)

An instance of CIM_Privilege is used to represent the privileges of the SM CLP Operator Role. Table 45 details the constraints for properties of CIM_Privilege beyond those specified in the *Role Based Authorization Profile* when it used to model the privileges of the SM CLP Operator Role.

Table 45 – Class: CIM_Privilege (Operator)

Properties and Methods	Requirement	Description
PrivilegeGranted	Mandatory	Matches TRUE
Activities	Mandatory	See section 7.5.3.2.1.
ActivityQualifiers	Mandatory	See section 7.5.3.2.2.
ElementName	Mandatory	Matches "SM CLP Operator Privilege"

10.23 CIM_Privilege (Read Only)

An instance of CIM_Privilege is used to represent the privileges of the SM CLP Read Only Role. Table 46 details the constraints for properties of CIM_Privilege beyond those specified in the *Role Based Authorization Profile* when it used to model the privileges of the SM CLP Read Only Role.

Table 46 – Class: CIM_Privilege (Read Only)

Properties and Methods	Requirement	Description
PrivilegeGranted	Mandatory	Matches TRUE
Activities	Mandatory	See section 7.5.4.2.1.
ActivityQualifiers	Mandatory	See section 7.5.4.2.2.
ElementName	Mandatory	Matches "SM CLP Read Only Privilege"

10.24 CIM_ProtocolService

CIM_ProtocolService represents the CLP service.

Table 47 – Class: CIM_ProtocolService

Properties	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
Name	Mandatory	**Key**
Protocol	Mandatory	See section 7.1.1.
MaxConnections	Mandatory	A value of 0 (zero) shall indicate unknown.
RequestedState	Mandatory	See section 7.1.3.
EnabledState	Mandatory	See section 7.1.3.
ElementName	Mandatory	See section 7.1.4.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
RequestStateChange( )	Conditional	See section 8.1.

10.25 CIM_ProvidesEndpoint

CIM_ProvidesEndpoint is used to associate the instance of CIM_ProtocolService with an instance of CIM_CLPProtocolEndpoint representing a session with the service.

Table 48 – Class: CIM_ProvidesEndpoint

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to the instance of CIM_ProtocolService. Cardinality 1
Dependent	Mandatory	**Key** This property shall be a reference to an instance of CIM_CLPProtocolEndpoint. Cardinality *

10.26 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Command Line Protocol Service Profile*.The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in Table 49, the behavior of the CIM_RegisteredProfile instance is in accordance with the constraints specified in the *Profile Registration Profile*.

Table 49 – Class: CIM_RegisteredProfile

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Command Line Protocol Service".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 ("DMTF").

NOTE: Previous versions of this document included the suffix “Profile” for the RegisteredName value. If implementations querying for the RegisteredName value find the suffix “Profile”, they should ignore the suffix, with any surrounding white spaces, before any comparison is done with the value as specified in this document.

10.27 CIM_Role (Administrator)

An instance of CIM_Role is used to model the CLP Administrator Role. Table 50 details the constraints for properties of CIM_Role beyond those specified in the *Role Based Authorization Profile* when it is used to model the CLP Administrator Role.

Table 50 – Class: CIM_Role (Administrator)

Properties and Methods	Requirement	Notes
RoleCharacteristics	Mandatory	Shall contain 2 (Static)
CommonName	Mandatory	Matches "CIM:CLP:Administrator"
ElementName	Mandatory	Matches "SM CLP Administrator"

10.28 CIM_Role (Operator)

An instance of CIM_Role is used to model the CLP Operator Role. Table 51 details the constraints for properties of CIM_Role beyond those specified in the *Role Based Authorization Profile* when it is used to model the CLP Operator Role.

Table 51 – Class: CIM_Role (Operator)

Properties and Methods	Requirement	Notes
RoleCharacteristics	Mandatory	Shall contain 2 (Static)
CommonName	Mandatory	Matches "CIM:CLP:Operator"
ElementName	Mandatory	Matches "SM CLP Operator"

10.29 CIM_Role (Read Only)

An instance of CIM_Role is used to model the CLP Read Only Role. Table 52 details the constraints for properties of CIM_Role beyond those specified in the *Role Based Authorization Profile* when it is used to model the CLP Read Only Role.

Table 52 – Class: CIM_Role (Read Only)

Properties and Methods	Requirement	Notes
RoleCharacteristics	Mandatory	Shall contain 2 (Static)
CommonName	Mandatory	Matches "CIM:CLP:Read Only"
ElementName	Mandatory	Matches "SM CLP Read Only"

10.30 CIM_RoleLimitedToTarget

Table 53 details the constraints for properties of CIM_RoleLimitedToTarget beyond those specified in the *Role Based Authorization Profile*.

Table 53 – Class: CIM_RoleLimitedToTarget

Properties and Methods	Requirement	Description
TargetElement	Mandatory	See section 7.5.1.1. Cardinality 1

10.31 CIM_ServiceAccessBySAP

CIM_ServiceAccessBySAP is used to associate the instance of CIM_ProtocolService with an instance of CIM_TCPProtocolEndpoint over which a session with the service can be established.

Table 54 – Class: CIM_ServiceAccessBySAP

Properties	Requirement	Notes
Antecedent	Mandatory	**Key** This property shall be a reference to the instance of CIM_ProtocolService. Cardinality 1..*
Dependent	Mandatory	**Key** This property shall be a reference to an instance of CIM_TCPProtocolEndpoint. Cardinality *

10.32 CIM_ServiceAffectsElement – AdminDomain

CIM_ServiceAffectsElement associates an instance of CIM_ProtocolService with an instance of CIM_AdminDomain representing the management domain of the CLP service.

Table 55 – Class: CIM_ServiceAffectsElement

Properties	Requirement	Notes
ServiceProvided	Mandatory	**Key** This property shall be a reference to the Central Instance of the profile. Cardinality 1
UserOfService	Mandatory	**Key** shall be a reference to the Central Instance defined in *DSP1007*. Cardinality 1
ElementAffects	Mandatory	Matches 5 (Manages)

10.33 CIM_ServiceAffectsElement – Job Queue

CIM_ServiceAffectsElement associates an instance of CIM_ProtocolService with the instance of CIM_JobQueue where CLP operations are executed.

Table 56 – Class: CIM_ServiceAffectsElement

Properties	Requirement	Notes
ServiceProvided	Mandatory	**Key** This property shall be a reference to the Central Instance of the profile. Cardinality 1
UserOfService	Mandatory	**Key** shall be a reference to CIM_JobQueue. Cardinality 1
ElementAffects	Mandatory	Shall contain the values 5 (Manages) and 6 (Consumes)

ANNEX A
(informative)

Change Log

Version	Date	Description
1.0.0a	10/10/2006	Preliminary Standard
1.0.0	06/17/2009	DMTF Standard Release