Document Number: DSP0818

Date: 2009-07-29

Version: 1.0.0

DHCP Client Profile SM CLP Command Mapping Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Tables

Foreword

The *DHCP Client Profile SM CLP Command Mapping Specification* (DSP0818) was prepared by the Server Management Working Group.

Conventions

The pseudo code conventions utilized in this document are the Recipe Conventions as defined in SNIA SMI-S 1.1.0, Section 7.6.

Acknowledgements

The authors wish to acknowledge the following participants from the DMTF Server Management Working Group:

Editor:

- Aaron Merkin – IBM

Contributors:

- Jon Hass – Dell
- Khachatur Papanyan – Dell
- Enoch Suen – Dell
- Jeff Hilland – HP
- Christina Shaw – HP
- Aaron Merkin – IBM
- Perry Vincent – Intel
- John Leung – Intel

Introduction

This document defines the SM CLP mapping for CIM elements described in the *DHCP Client Profile*. The information in this specification, combined with *SM CLP-to-CIM Common Mapping Specification V1.0* (DSP0216), is intended to be sufficient to implement SM CLP commands relevant to the classes, properties and methods described in the *DHCP Client Profile* using CIM operations.

The target audience for this specification is implementers of the SM CLP support for the *DHCP Client Profile*.

DHCP Client Profile SM CLP Command Mapping Specification

1 Scope

This specification contains the requirements for an implementation of the SM CLP to provide access to, and implement the behaviors of, the *DHCP Client Profile*.

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 DSP0216, *SM CLP-to-CIM Common Mapping Specification 1.0*, http://www.dmtf.org/standards/published_documents/DSP0216_1.0.pdf

DMTF DSP1037, *DHCP Client Profile 1.0*, http://www.dmtf.org/standards/published_documents/DSP1037_1.0.pdf

SNIA, *Storage Management Initiative Specification (SMI-S) 1.1.0*, November 2005, http://www.snia.org/tech_activities/standards/curr_standards/smi/

2.2 Other References

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*

3 Terms and Definitions

For the purposes of this document, the following terms and definitions 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

shall

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

3.9

shall not

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

3.10

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

3.11

should not

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

3.12

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

Common Information Model

4.3
DMTF
Distributed Management Task Force

4.4
IETF
Internet Engineering Task Force

4.5
RFC
Request for Comment

4.6
SM
Server Management

4.7
SMI-S
Storage Management Initiative Specification

4.8
SNIA
Storage Networking Industry Association

5 Recipes

The following is a list of the common recipes used by the mappings in this specification. For a definition of each recipe, see *SM CLP-to-CIM Common Mapping Specification 1.0* (DSP0216).

- smStartRSC()
- smStopRSC()
- smResetRSC()
- smShowInstance()
- smShowInstances()
- smSetInstance()
- smShowAssociationInstances()
- smShowAssociationInstance()

This mapping does not define any recipes for local reuse.

6 Mappings

The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in the *DHCP Client Profile*. Requirements specified here related to support for a CLP verb for a particular class are solely within the context of this profile.

6.1 CIM_DHCPCapabilities

The `cd` and `help` verbs shall supported as described in DSP0216.

Table 1 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 1 is for informational purposes only; in case of a conflict between this table and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 1.

Table 1 – Command Verb Requirements for CIM_DHCPCapabilities

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.1.2.
Start	Not supported	
Stop	Not supported	

No mapping is defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.1.1 Ordering of Results

When results are returned for multiple instances of CIM_DHCPCapabilities, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_DHCPCapabilities are unordered; therefore, no algorithm is defined.

6.1.2 Show

The `show` verb is used to display information about instances of CIM_DHCPCapabilities. Implementations shall support the use of the `show` verb with CIM_DHCPCapabilities.

6.1.2.1 Show Command Form for Single Instance

This command form is used when the `show` verb applies to a single instance of CIM_DHCPCapabilities.

6.1.2.1.1 Command Form

```
show <CIM_DHCPCapabilities single instance>
```

6.1.2.1.2 CIM Requirements

See CIM_DHCPCapabilities in the "CIM Elements" section of the *DHCP Client Profile* for the list of mandatory properties.

6.1.2.1.3 Behavior Requirements

6.1.2.1.3.1 Preconditions

`#all` is true if the "-all" option was specified with the command; otherwise, `#all` is false.

6.1.2.1.3.2 Pseudo Code

```
$instance=<CIM_DHCPCapabilities single instance>
   #propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = {//all mandatory non-key properties};
   }
&smShowInstance($instance.getObjectPath(), #propertylist[]);
&smEnd;
```

6.1.2.2 Show Command Form for Multiple Instances

This command form is used when the show verb applies to multiple instances of CIM_DHCPCapabilities. This command form corresponds to UFsT-based selection within a capabilities collection.

6.1.2.2.1 Command Form

```
show <CIM_DHCPCapabilities multiple instances>
```

6.1.2.2.2 CIM Requirements

See CIM_DHCPCapabilities in the "CIM Elements" section of the *DHCP Client Profile* for the list of mandatory properties.

6.1.2.2.3 Behavior Requirements

6.1.2.2.3.1 Preconditions

$containerInstance contains the instance of CIM_ConcreteCollection for which contained CIM_Capabilities instances are displayed. CIM_Capabilities instances are addressed via the aggregating instance of CIM_ConcreteCollection

#all is true if the "-all" option was specified with the command; otherwise, #all is false.

6.1.2.2.3.2 Pseudo Code

```
#propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = {//all non-key properties};
   }
&smShowInstances ( "CIM_DHCPCapabilities", "CIM_MemberOfCollection",
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.2 CIM_DHCPSettingData

The cd and help verbs shall be supported as described in DSP0216.

Table 2 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 2 is for informational purposes only; in case of a conflict between Table 2 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 2.

Table 2 – Command Verb Requirements for CIM_DHCPSettingData

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.2.2.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.2.1 Ordering of Results

When results are returned for multiple instances of CIM_DHCPSettingData, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_DHCPSettingData are unordered; therefore, no algorithm is defined.

6.2.2 Show

The `show` verb is used to display information about instances of CIM_DHCPSettingData. Implementations shall support the use of the `show` verb with CIM_USBRedirectionService.

6.2.2.1 Show Command Form for Single Instance

This command form is used when the `show` verb applies to a single instance of CIM_DHCPSettingData.

6.2.2.1.1 Command Form

`show <CIM_DHCPSettingData` *`single instance`*`>`

6.2.2.1.2 CIM Requirements

See CIM_DHCPSettingData in the "CIM Elements" section of the *DHCP Client Profile* for the list of mandatory properties.

6.2.2.1.3 Behavior Requirements

6.2.2.1.3.1 Preconditions

`#all` is true if the "-all" option was specified with the command; otherwise, `#all` is false.

6.2.2.1.3.2 Pseudo Code

```
$instance=<CIM_DHCPSettingData single instance>
   #propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = { //all mandatory non-key properties }
   }
&smShowInstance($instance.getObjectPath(), #propertylist[]);
&smEnd;
```

6.2.2.2 Show Command Form for Multiple Instances

This command form is used when the `show` verb applies to multiple instances of CIM_DHCPSettingData. This command form corresponds to UFsT-based selection within a scoping system.

6.2.2.2.1 Command Form

```
show <CIM_DHCPSettingData multiple instances>
```

6.2.2.2.2 CIM Requirements

See CIM_DHCPSettingData in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.2.2.2.3 Behavior Requirements

6.2.2.2.3.1 Preconditions

`$containerInstance` represents the instance of CIM_IPAssignmentSettingData for which related CIM_DHCPSettingData instances are displayed.

`#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

6.2.2.2.3.2 Pseudo Code

```
#propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = { //all mandatory non-key properties }
   }
&smShowInstances ( “CIM_DHCPSettingData”, “CIM_OrderedComponent",
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.3 CIM_DHCPProtocolEndpoint

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 3 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 3 is for informational purposes only; in case of a conflict between Table 3 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 3.

Table 3 – Command Verb Requirements for CIM_DHCPProtocolEndpoint

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	May	See 6.3.2.
Show	Shall	See 6.3.3.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `start`, and `stop`.

6.3.1 Ordering of Results

When results are returned for multiple instances of CIM_DHCPProtocolEndpoint, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_DHCPProtocolEndpoint are unordered; therefore, no algorithm is defined.

6.3.2 Set

The `set` verb is used to modify descriptive properties of the CIM_DHCPProtocolEndpoint instance. Implementations may support the use of the `set` verb with CIM_DHCPProtocolEndpoint.

6.3.2.1 General Usage of Set for a Single Property

This command form corresponds to the general usage of the `set` verb to modify a single property of a target instance. This is the most common case.

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *DHCP Client Profile*.

6.3.2.1.1 Command Form

```
set <CIM_DHCPProtocolEndpoint single object> <propertyname>=<propertyvalue>
```

6.3.2.1.2 CIM Requirements

See CIM_DHCPProtocolEndpoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of any modifiable properties.

6.3.2.1.3 Behavior Requirements

```
$instance=<CIM_DHCPProtocolEndpoint single object>
   #propertyNames[] = {<propertyname>};
   #propertyValues[] = {<propertyvalue>};
&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.3.2.2 General Usage of Set for Multiple Properties

This command form corresponds to the general usage of the `set` verb to modify multiple properties of a target instance where there is not an explicit relationship between the properties. This is the most common case.

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the DHCP Client Profile.

6.3.2.2.1 Command Form

```
set <CIM_DHCPProtocolEndpoint multiple objects> <propertyname1>=<propertyvalue1>
   <propertynamen>=<propertyvaluen>
```

6.3.2.2.2 CIM Requirements

See CIM_DHCPProtocolEndpoint in the “CIM Elements” section of the DHCP Client Profile for the list of mandatory properties.

6.3.2.2.3 Behavior Requirements

```
$instance=<CIM_DHCPProtocolEndpoint multiple objects>
#propertyNames[] = {<propertyname>};
   for #i < n
   {
      #propertyNames[#i] = <propertname#i>
      #propertyValues[#i] = <propertyvalue#i>
   }
&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.3.3 Show

The `show` verb is used to display information about instances of CIM_DHCPProtocolEndpoint. Implementations shall support the use of the `show` verb with CIM_DHCPProtocolEndpoint.

6.3.3.1 Show Command Form for Single Instance

This command form is used when the `show` verb applies to a single instance of CIM_DHCPProtocolEndpoint.

6.3.3.1.1 Command Form

```
show <CIM_DHCPProtocolEndpoint single instance>
```

6.3.3.1.2 CIM Requirements

See CIM_DHCPProtocolEndpoint in the “CIM Elements” section of the DHCP Client Profile for the list of mandatory properties.

6.3.3.1.3 Behavior Requirements

6.3.3.1.3.1 Preconditions

`#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

6.3.3.1.3.2 Pseudo Code

```
$instance=<CIM_DHCPProtocolEndpoint single instance>
   #propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = { //all mandatory non-key properties };
   }
&smShowInstance($instance.getObjectPath(), #propertylist[]);
&smEnd;
```

6.3.3.2 Show Command Form for Multiple Instances

This command form is used when the `show` verb applies to multiple instances of CIM_DHCPProtocolEndpoint. This command form corresponds to UFsT-based selection within a scoping system.

6.3.3.2.1 Command Form

```
show <CIM_DHCPProtocolEndpoint multiple instances>
```

6.3.3.2.2 CIM Requirements

See CIM_DHCPProtocolEndpoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.3.3.2.3 Behavior Requirements

6.3.3.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ComputerSystem for which scoped CIM_DHCPProtocolEndpoint instances are displayed. The *DHCP Client Profile* requires that the CIM_DHCPProtocolEndpoint instance be associated with its scoping system via an instance of the CIM_HostedAccessPoint association.

`#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

6.3.3.2.3.2 Pseudo Code

```
#propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = { //all mandatory non-key properties };
   }
&smShowInstances ( “CIM_DHCPProtocolEndpoint”, “CIM_HostedAccessPoint",
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.4 CIM_ElementCapabilities

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 4 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 4 is for informational purposes only; in case of a conflict between Table 4 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 4.

Table 4 – Command Verb Requirements for CIM_ElementCapabilities

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.4.2.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.4.1 Ordering of Results

When results are returned for multiple instances of CIM_ElementCapabilities, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_ElementCapabilities are unordered; therefore, no algorithm is defined.

6.4.2 Show

The `show` verb is used to display information about instances of CIM_ElementCapabilities. Implementations shall support the use of the `show` verb with CIM_ElementCapabilities.

6.4.2.1 Show Command Form for Multiple Instances – CIM_DHCPCapabilities Reference

This command form is used when the `show` verb applies to multiple instances. This command form corresponds to a `show` command issued against instances of CIM_ElementCapabilities where only one reference is specified and the reference is to an instance of CIM_DHCPCapabilities.

6.4.2.1.1 Command Form

```
show <CIM_ElementCapabilities multiple instances>
```

6.4.2.1.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.4.2.1.3 Behavior Requirements

6.4.2.1.3.1 Preconditions

`$instance` contains the instance of CIM_DHCPCapabilities which is referenced by CIM_ElementCapabilities.

6.4.2.1.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_ElementCapabilities”, $instance.getObjectPath() );
&smEnd;
```

6.4.2.2 Show Command Form for Single Instance – CIM_DHCPProtocolEndpoint Reference

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_ElementCapabilities where the reference specified is to an instance of CIM_DHCPProtocolEndpoint. An instance of CIM_DHCPProtocolEndpoiint is referenced by exactly one instance of CIM_ElementCapabilities; therefore, a single instance will be returned.

6.4.2.2.1 Command Form

```
show <CIM_ElementCapabilities single instance>
```

6.4.2.2.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.4.2.2.3 Behavior Requirements

6.4.2.2.3.1 Preconditions

`$instance` represents the instance of CIM_DHCPProtocolEndpoint, which is referenced by CIM_ElementCapabilities.

6.4.2.2.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_ElementCapabilities”, $instance.getObjectPath() );
&smEnd;
```

6.4.2.3 Show Command Form for Single Instance – Both References

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_ElementCapabilities where both references are specified; therefore, the desired instance is unambiguously identified.

6.4.2.3.1 Command Form

```
show <CIM_ElementCapabilities single instance>
```

6.4.2.3.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.4.2.3.3 Behavior Requirements

6.4.2.3.3.1 Preconditions

`$instanceA` contains the instance of CIM_DHCPCapabilities which is referenced by CIM_ElementCapabilities.

`$instanceB` contains the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_ElementCapabilities.

6.4.2.3.3.2 Pseudo Code

```
&smShowAssociationInstance ( “CIM_ElementCapabilities”, $instanceA.getObjectPath(),
    $instanceB.getObjectPath() );
&smEnd;
```

6.5 CIM_ElementSettingData

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 5 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 5 is for informational purposes only; in case of a conflict between Table 5 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 5.

Table 5 – Command Verb Requirements for CIM_ElementSettingData

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.5.2.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.5.1 Ordering of Results

When results are returned for multiple instances of CIM_ElementSettingData, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_ElementSettingData are unordered; therefore, no algorithm is defined.

6.5.2 Show

The `show` verb is used to display information about instances of CIM_ElementSettingData. Implementations shall support the use of the `show` verb with CIM_ElementSettingData.

6.5.2.1 Show Command Form for Multiple Instances – CIM_DHCPProtocolEndpoint Reference

This command form is used when the `show` verb applies to multiple instances. This command form corresponds to a `show` command issued against instances of CIM_ElementSettingData where only one reference is specified and the reference is to an instance of CIM_DHCPProtocolEndpoint.

6.5.2.1.1 Command Form

```
show <CIM_ElementSettingData multiple instances>
```

6.5.2.1.2 CIM Requirements

See CIM_ElementSettingData in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.5.2.1.3 Behavior Requirements

6.5.2.1.3.1 Preconditions

`$instance` contains the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_ElementSettingData.

`#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

6.5.2.1.3.2 Pseudo Code

```
#propertylist[] = null;
if (#all == false)
   {
      #propertylist[] = { "IsCurrent" };
   }
&smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getObjectPath(),
   #propertylist[] );
&smEnd;
```

6.5.2.2 Show Command Form for Multiple Instances – CIM_DHCPSettingData Reference

This command form is used when the `show` verb applies to multiple instances. This command form corresponds to a `show` command issued against instances of CIM_ElementSettingData where only one reference is specified and the reference is to an instance of CIM_DHCPSettingData.

6.5.2.2.1 Command Form

```
show <CIM_ElementSettingData multiple instances>
```

6.5.2.2.2 CIM Requirements

See CIM_ElementSettingData in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.5.2.2.3 Behavior Requirements

6.5.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_DHCPSettingData which is referenced by CIM_ElementSettingData.

`#all` is true if the "-all" option was specified with the command; otherwise, `#all` is false.

6.5.2.2.3.2 Pseudo Code

```
#propertylist[] = null;
if (#all == false)
   {
      #propertylist[] = { "IsCurrent" };
   }
&smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getObjectPath(),
   #propertylist[] );
&smEnd;
```

6.5.2.3 Show Command Form for Single Instance – Both References

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_ElementSettingData where both references are specified; therefore, the desired instance is unambiguously identified.

6.5.2.3.1 Command Form

```
show <CIM_ElementSettingData single instance>
```

6.5.2.3.2 CIM Requirements

See CIM_ElementSettingData in the "CIM Elements" section of the *DHCP Client Profile* for the list of mandatory properties.

6.5.2.3.3 Behavior Requirements

6.5.2.3.3.1 Preconditions

`$instanceA` contains the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_ElementSettingData.

`$instanceB` contains the instance of CIM_DHCPSettingData which is referenced by CIM_ElementSettingData.

`#all` is true if the "-all" option was specified with the command; otherwise, `#all` is false.

6.5.2.3.3.2 Pseudo Code

```
#propertylist[] = null;
if (#all == false)
   {
      #propertylist[] = { "IsCurrent" };
   }
&smShowAssociationInstance ( “CIM_ElementSettingData”, $instanceA.getObjectPath(),
   $instanceB.getObjectPath(), #propertylist[] );
&smEnd;
```

6.6 CIM_HostedAccessPoint

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 6 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 6 is for informational purposes only; in case of a conflict between Table 6 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 6.

Table 6 – Command Verb Requirements for CIM_HostedAccessPoint

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.6.2.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.6.1 Ordering of Results

When results are returned for multiple instances of CIM_HostedAccessPoint, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_HostedAccessPoint are unordered; therefore, no algorithm is defined.

6.6.2 Show

The `show` verb is used to display information about instances of CIM_HostedAccessPoint. Implementations shall support the use of the `show` verb with CIM_HostedAccessPoint.

6.6.2.1 Show Command Form for Multiple Instances – CIM_ComputerSystem Reference

This command form is used when the `show` verb applies to multiple instances. This command form corresponds to a `show` command issued against instances of CIM_HostedAccessPoint where only one reference is specified and the reference is to an instance of CIM_ComputerSystem.

6.6.2.1.1 Command Form

```
show <CIM_HostedAccessPoint multiple instances>
```

6.6.2.1.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.6.2.1.3 Behavior Requirements

6.6.2.1.3.1 Preconditions

`$instance` contains the instance of CIM_ComputerSystem which is referenced by CIM_HostedAccessPoint.

6.6.2.1.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_HostedAccessPoint”, $instance.getObjectPath() );
&smEnd;
```

6.6.2.2 Show Command Form for Single Instance – CIM_DHCPProtocolEndpoint Reference

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_HostedAccessPoint where the reference specified is to an instance of CIM_DHCPProtocolEndpoint. An instance of CIM_DHCPProtocolEndpoint is referenced by exactly one instance of CIM_HostedAccessPoint; therefore, a single instance will be returned.

6.6.2.2.1 Command Form

```
show <CIM_HostedAccessPoint single instance>
```

6.6.2.2.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.6.2.2.3 Behavior Requirements

6.6.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_HostedAccessPoint.

6.6.2.2.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_HostedAccessPoint”, $instance.getObjectPath() );
&smEnd;
```

6.6.2.3 Show Command Form for Single Instance – CIM_RemoteServiceAccessPoint Reference

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against CIM_HostedAccessPoint where the reference specified is to an instance of CIM_RemoteServiceAccessPoint. An instance of CIM_RemoteServiceAccessPoint is referenced by exactly one instance of CIM_HostedAccessPoint; therefore, a single instance will be returned.

6.6.2.3.1 Command Form

```
show <CIM_HostedAccessPoint single instance>
```

6.6.2.3.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.6.2.3.3 Behavior Requirements

6.6.2.3.3.1 Preconditions

`$instance` contains the instance of CIM_RemoteServiceAccessPoint which is referenced by CIM_HostedAccessPoint.

6.6.2.3.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_HostedAccessPoint”, $instance.getObjectPath() );
&smEnd;
```

6.6.2.4 Show Command Form for Single Instance – Both References (DHCPProtocolEndpoint)

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_HostedAccessPoint where both references are specified; therefore, the desired instance is unambiguously identified.

6.6.2.4.1 Command Form

```
show <CIM_HostedAccessPoint single instance>
```

6.6.2.4.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.6.2.4.3 Behavior Requirements

6.6.2.4.3.1 Preconditions

`$instanceA` contains the instance of CIM_ComputerSystem which is referenced by CIM_HostedAccessPoint.

`$instanceB` contains the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_HostedAccessPoint.

6.6.2.4.3.2 Pseudo Code

```
&smShowAssociationInstance ( “CIM_HostedAccessPoint”, $instanceA.getObjectPath(),
   $instanceB.getObjectPath() );
&smEnd;
```

6.6.2.5 Show Command Form for Single Instance – Both References (RemoteServiceAccessPoint)

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_HostedAccessPoint where both references are specified; therefore, the desired instance is unambiguously identified.

6.6.2.5.1 Command Form

```
show <CIM_HostedAccessPoint single instance>
```

6.6.2.5.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.6.2.5.3 Behavior Requirements

6.6.2.5.3.1 Preconditions

`$instanceA` contains the instance of CIM_ComputerSystem which is referenced by CIM_HostedAccessPoint.

`$instanceB` contains the instance of CIM_RemoteServiceAccessPoint which is referenced by CIM_HostedAccessPoint.

6.6.2.5.3.2 Pseudo Code

```
&smShowAssociationInstance ( “CIM_HostedAccessPoint”, $instanceA.getObjectPath(),
    $instanceB.getObjectPath() );
&smEnd;
```

6.7 CIM_RemoteAccessAvailableToElement

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 7 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 7 is for informational purposes only; in case of a conflict between Table 7 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 7.

Table 7 – Command Verb Requirements for CIM_RemoteAccessAvailableToElement

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.7.2.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.7.1 Ordering of Results

When results are returned for multiple instances of CIM_RemoteAccessAvailableToElement, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_RemoteAccessAvailableToElement are unordered; therefore, no algorithm is defined.

6.7.2 Show

The show verb is used to display information about instances of CIM_RemoteAccessAvailableToElement. Implementations shall support the use of the show verb with CIM_RemoteAccessAvailableToElement.

6.7.2.1 Show Command Form for Multiple Instances – CIM_RemoteServiceAccessPoint Reference

This command form is used when the show verb applies to multiple instances. This command form corresponds to a show command issued against instances of CIM_RemoteAccessAvailableToElement where only one reference is specified and the reference is to an instance of CIM_RemoteServiceAccessPoint.

6.7.2.1.1 Command Form

```
show <CIM_RemoteAccessAvailableToElement multiple instances>
```

6.7.2.1.2 CIM Requirements

See CIM_RemoteAccessAvailableToElement in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.7.2.1.3 Behavior Requirements

6.7.2.1.3.1 Preconditions

$instance contains the instance of CIM_RemoteServiceAccessPoint which is referenced by CIM_RemoteAccessAvailableToElement.

There is only a single property and it is always returned.

6.7.2.1.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_RemoteAccessAvailableToElement”,
    $instance.getObjectPath(), null );
&smEnd;
```

6.7.2.2 Show Command Form for Multiple Instances – CIM_DHCPProtocolEndpoint Reference

This command form is used when the show verb applies to multiple instances. This command form corresponds to a show command issued against instances of CIM_RemoteAccessAvailableToElement where the reference specified is to an instance of CIM_DHCPProtocolEndpoint. The *DHCP Client Profile* allows the implementation to model the DHCP servers discovered by the client in addition to the DHCP Service that actually provides the configuration; therefore, it is possible for there to be multiple CIM_RemoteAccessAvailableToElement associations that reference the CIM_DHCPProtocolEndpoint instance.

6.7.2.2.1 Command Form

```
show <CIM_RemoteAccessAvailableToElement multiple instances>
```

6.7.2.2.2 CIM Requirements

See CIM_RemoteAccessAvailableToElement in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.7.2.2.3 Behavior Requirements

6.7.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_RemoteAccessAvailableToElement.

There is only a single property and it is always returned.

6.7.2.2.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_RemoteAccessAvailableToElement”,
   $instance.getObjectPath(), null );
&smEnd;
```

6.7.2.3 Show Command Form for Single Instance – Both References

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_RemoteAccessAvailableToElement where both references are specified; therefore, the desired instance is unambiguously identified.

6.7.2.3.1 Command Form

```
show <CIM_RemoteAccessAvailableToElement single instance>
```

6.7.2.3.2 CIM Requirements

See CIM_RemoteAccessAvailableToElement in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.7.2.3.3 Behavior Requirements

6.7.2.3.3.1 Preconditions

`$instanceA` represents the referenced instance of CIM_RemoteServiceAccessPoint which is referenced by CIM_RemoteAccessAvailableToElement.

`$instanceB` represents the referenced instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_RemoteAccessAvailableToElement.

There is only a single property and it is always returned.

6.7.2.3.3.2 Pseudo Code

```
&smShowAssociationInstance ( “CIM_RemoteAccessAvailableToElement”,
   $instanceA.getObjectPath(), $instanceB.getObjectPath(), null );
&smEnd;
```

6.8 CIM_RemoteServiceAccessPoint

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 8 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 8 is for informational purposes only; in case of a conflict between Table 8 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 8.

Table 8 – Command Verb Requirements for CIM_RemoteServiceAccessPoint

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	May	See 6.8.2.
Show	Shall	See 6.8.3.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `start`, and `stop`.

6.8.1 Ordering of Results

When results are returned for multiple instances of CIM_RemoteServiceAccessPoint, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_RemoteServiceAccessPoint are unordered; therefore, no algorithm is defined.

6.8.2 Set

The `set` verb is used to modify descriptive properties of the CIM_RemoteServiceAccessPoint instance. Implementations may support the use of the `set` verb with CIM_RemoteServiceAccessPoint.

6.8.2.1 General Usage of Set for a Single Property

This command form corresponds to the general usage of the `set` verb to modify a single property of a target instance. This is the most common case.

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *DHCP Client Profile*.

6.8.2.1.1 Command Form

```
set <CIM_RemoteServiceAccessPoint single object> <propertyname>=<propertyvalue>
```

6.8.2.1.2 CIM Requirements

See CIM_RemoteServiceAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of modifiable properties.

6.8.2.1.3 Behavior Requirements

```
$instance=<CIM_RemoteServiceAccessPoint single object>
   #propertyNames[] = {<propertyname>};
   #propertyValues[] = {<propertyvalue>};
&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.8.2.2 General Usage of Set for Multiple Properties

This command form corresponds to the general usage of the `set` verb to modify multiple properties of a target instance where there is not an explicit relationship between the properties. This is the most common case.

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *DHCP Client Profile*.

6.8.2.2.1 Command Form

```
set <CIM_RemoteServiceAccessPoint multiple objects> <propertyname1>=<propertyvalue1>
   <propertynamen>=<propertyvaluen>
```

6.8.2.2.2 CIM Requirements

See CIM_RemoteServiceAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of modifiable properties.

6.8.2.2.3 Behavior Requirements

```
$instance=<CIM_RemoteServiceAccessPoint multiple objects>
   #propertyNames[] = {<propertyname>};
   for #i < n
   {
      #propertyNames[#i] = <propertname#i>
      #propertyValues[#i] = <propertyvalue#i>
   }
&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.8.3 Show

The `show` verb is used to display information about instances of CIM_RemoteServiceAccessPoint. Implementations shall support the use of the `show` verb with CIM_RemoteServiceAccessPoint.

6.8.3.1 Show Command Form for Single Instance

This command form is used when the `show` verb applies to a single instance of CIM_RemoteServiceAccessPoint.

6.8.3.1.1 Command Form

```
show <CIM_RemoteServiceAccessPoint single instance>
```

6.8.3.1.2 CIM Requirements

See CIM_RemoteServiceAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.8.3.1.3 Behavior Requirements

6.8.3.1.3.1 Preconditions

`#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

6.8.3.1.3.2 Pseudo Code

```
$instance=<CIM_RemoteServiceAccessPoint single instance>
   #propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = {“AccessContext”, “AccessInfo”, “InfoFormat”, “ElementName”};
   }
&smShowInstance($instance.getObjectPath(), #propertylist[]);
&smEnd;
```

6.8.3.2 Show Command Form for Multiple Instances

This command form is used when the `show` verb applies to multiple instances of CIM_RemoteServiceAccessPoint. This command form corresponds to UFsT-based selection within a scoping system.

6.8.3.2.1 Command Form

```
show <CIM_RemoteServiceAccessPoint multiple instances>
```

6.8.3.2.2 CIM Requirements

See CIM_RemoteServiceAccessPoint in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.8.3.2.3 Behavior Requirements

6.8.3.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ComputerSystem for which scoped CIM_RemoteServiceAccessPoint instances are displayed. The *DHCP Client Profile* requires that the CIM_RemoteServiceAccessPoint instance be associated with its scoping system via an instance of the CIM_HostedAccessPoint association.

`#all` is true if the “-all” option was specified with the command; otherwise, `#all` is false.

6.8.3.2.3.2 Pseudo Code

```
#propertylist[] = null;
   if (false == #all)
   {
      #propertylist[] = {“AccessContext”, “AccessInfo”, “InfoFormat”, “ElementName”};
   }
&smShowInstances ( “CIM_RemoteServiceAccessPoint”, “CIM_HostedAccessPoint",
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.9 CIM_SAPSAPDependency

The `cd` and `help` verbs shall be supported as described in DSP0216.

Table 9 lists each SM CLP verb, the required level of support for the verb in conjunction with the target class, and when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 9 is for informational purposes only; in case of a conflict between Table 9 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 9.

Table 9 – Command Verb Requirements for CIM_SAPSAPDependency

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	
Set	Not supported	
Show	Shall	See 6.9.2.
Start	Not supported	
Stop	Not supported	

No mappings are defined for the following verbs for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.9.1 Ordering of Results

When results are returned for multiple instances of CIM_SAPSAPDependency, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering.

- Results for CIM_SAPSAPDependency are unordered; therefore, no algorithm is defined.

6.9.2 Show

The `show` verb is used to display information about instances of CIM_SAPSAPDependency. Implementations shall support the use of the `show` verb with CIM_SAPSAPDependency.

6.9.2.1 Show Command Form for Single Instance – CIM_IPProtocolEndpoint Reference

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_SAPSAPDependency where only one reference is specified and the reference is to an instance of CIM_IPProtocolEndpoint.

6.9.2.1.1 Command Form

```
show <CIM_SAPSAPDependency single instance>
```

6.9.2.1.2 CIM Requirements

See CIM_SAPSAPDependency in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.9.2.1.3 Behavior Requirements

6.9.2.1.3.1 Preconditions

`$instance` represents the instance of CIM_IPProtocolEndpoint, which is referenced by CIM_SAPSAPDependency.

6.9.2.1.3.2 Pseudo Code

```
&smShowAssociationInstances ("CIM_SAPSAPDependency", $instance.getObjectPath() );
&smEnd;
```

6.9.2.2 Show Command Form for Single Instance – CIM_DHCPProtocolEndpoint Reference

This command form is used when the `show` verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_SAPSAPDependency where the reference specified is to an instance of CIM_DHCPProtocolEndpoint. An instance of CIM_DHCPProtocolEndpoiint is referenced by exactly one instance of CIM_SAPSAPDependency; therefore, a single instance will be returned.

6.9.2.2.1 Command Form

```
show <CIM_SAPSAPDependency single instance>
```

6.9.2.2.2 CIM Requirements

See CIM_SAPSAPDependency in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.9.2.2.3 Behavior Requirements

6.9.2.2.3.1 Preconditions

`$instance` represents the instance of CIM_DHCPProtocolEndpoint, which is referenced by CIM_SAPSAPDependency.

6.9.2.2.3.2 Pseudo Code

```
&smShowAssociationInstances ("CIM_SAPSAPDependency", $instance.getObjectPath() );
&smEnd;
```

6.9.2.3 Show Command Form for Single Instance – Both References

This command form is used when the show verb applies to a single instance. This command form corresponds to a `show` command issued against instances of CIM_SAPSAPDependency where both references are specified; therefore, the desired instance is unambiguously identified.

6.9.2.3.1 Command Form

```
show <CIM_SAPSAPDependency single instance>
```

6.9.2.3.2 CIM Requirements

See CIM_SAPSAPDependency in the “CIM Elements” section of the *DHCP Client Profile* for the list of mandatory properties.

6.9.2.3.3 Behavior Requirements

6.9.2.3.3.1 Preconditions

`$instanceA` represents the instance of CIM_IPProtocolEndpoint which is referenced by CIM_SAPSAPDependency.

`$instanceB` represents the instance of CIM_DHCPProtocolEndpoint which is referenced by CIM_SAPSAPDependency.

6.9.2.3.3.2 Pseudo Code

```
&smShowAssociationInstance ( “CIM_SAPSAPDependency”, $instanceA.getObjectPath(),
    $instanceB.getObjectPath() );
&smEnd;
```

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.0.0	07/29/09		DMTF Standard release.