Document Number: DSP0819

Date: 2009-07-14

Version: 1.0.0

DNS Client Profile SM CLP Command Mapping Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Tables

65

Foreword

The *DNS Client Profile SM CLP Command Mapping Specification* (DSP0819) was prepared by the Server Management Working Group.

Conventions

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

Acknowledgements

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

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

Introduction

This document defines the SM CLP mapping for CIM elements described in the *DNS Client Profile*. The information in this specification, combined with the *SM CLP-to-CIM Common Mapping Specification 1.0*, is intended to be sufficient to implement SM CLP commands relevant to the classes, properties, and methods described in the *DNS Client Profile* using CIM operations.

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

DNS 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 *DNS 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 DSP1038, *DNS Client Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1038_1.0.pdf

SNIA, *Storage Management Initiative Specification (SMI-S) 1.1.0*,
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*,
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.

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

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
DMTF
Distributed Management Task Force

4.4
IETF
Internet Engineering Task Force

4.5
SM
Server Management

4.6
SMI-S
Storage Management Initiative Specification

4.7
SNIA
Storage Networking Industry Association

4.8
UFsT
User Friendly selection Tag

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).

- 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 *DNS 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_DNSGeneralSettingData

The `cd` and `help` verbs shall be 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 Table 1 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_DNSGeneralSettingData

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	

Command Verb	Requirement	Comments
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.1.2.
show	Shall	See 6.1.3.
start	Not supported	
stop	Not supported	

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

6.1.1 Ordering of Results

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

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

6.1.2 Set

This section describes how to implement the `set` verb when it is applied to an instance of CIM_DNSGeneralSettingData. Implementations may support the use of the `set` verb with CIM_DNSGeneralSettingData.

The `set` verb is used to modify the properties of the CIM_DNSGeneralSettingData instance.

6.1.2.1 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 *DNS Client Profile*.

6.1.2.1.1 Command Form

```
set <CIM_DNSGeneralSettingData single instance> <propertyname>=<propertyvalue>
```

6.1.2.1.2 CIM Requirements

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

6.1.2.1.3 Behavior Requirements

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

6.1.2.2 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 *DNS Client Profile*.

6.1.2.2.1 Command Form

```
set <CIM_DNSGeneralSettingData single instance> <propertyname1>=<propertyvalue1>
   <propertynamen>=<propertyvaluen>
```

6.1.2.2.2 CIM Requirements

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

6.1.2.2.3 Behavior Requirements

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

6.1.3 Show

This section describes how to implement the `show` verb when applied to an instance of CIM_DNSGeneralSettingData. Implementations shall support the use of the `show` verb with CIM_DNSGeneralSettingData.

The `show` verb is used to display information about CIM_DNSGeneralSettingData.

6.1.3.1 Show a Single Instance

This command form is for the `show` verb applied to a single instance of CIM_DNSGeneralSettingData.

6.1.3.1.1 Command Form

```
show <CIM_DNSGeneralSettingData single instance>
```

6.1.3.1.2 CIM Requirements

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

6.1.3.1.3 Behavior Requirements

6.1.3.1.3.1 Preconditions

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

6.1.3.1.3.2 Pseudo Code

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

6.1.3.2 Show Multiple Instances in Settings Collection

This command form is for the `show` verb applied to multiple instances of CIM_DNSGeneralSettingData. This command form corresponds to UFsT-based selection within a scoping system.

6.1.3.2.1 Command Form

```
show <CIM_DNSGeneralSettingData multiple instances>
```

6.1.3.2.2 CIM Requirements

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

6.1.3.2.3 Behavior Requirements

6.1.3.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ConcreteCollection for which related CIM_DNSGeneralSettingData instances are displayed. The CIM_DNSGeneralSettingData instance is associated with an instance of CIM_ConcreteCollection via an instance of the CIM_MemberOfCollection association.

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

6.1.3.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties }
   }
&smShowInstances ( “CIM_DNSGeneralSettingData”, “CIM_MemberOfCollection”,
   $containerInstance.getInstancePath(), #propertylist[] );
&smEnd;
```

6.1.3.3 Show Multiple Instances

This command form is for the `show` verb applied to multiple instances of CIM_DNSGeneralSettingData. This command form corresponds to UFsT-based selection within a scoping system.

6.1.3.3.1 Command Form

```
show <CIM_DNSGeneralSettingData multiple instances>
```

6.1.3.3.2 CIM Requirements

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

6.1.3.3.3 Behavior Requirements

6.1.3.3.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ConcreteCollection for which we are displaying related CIM_DNSGeneralSettingData instances. SM ME Addressing requires that the CIM_DNSGeneralSettingData instance be associated with an instance of CIM_ConcreteCollection via an instance of the CIM_MemberOfCollection association.

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

6.1.3.3.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties }
   }
&smShowInstances ( “CIM_DNSGeneralSettingData”, “CIM_MemberOfCollection”,
   $containerInstance.getInstancePath(), #propertylist[] );
&smEnd;
```

6.2 CIM_DNSSettingData

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 instances of 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_DNSSettingData

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.2.2.
show	Shall	See 6.2.3.
start	Not supported	
stop	Not supported	

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

6.2.1 Ordering of Results

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

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

6.2.2 Set

This section describes how to implement the `set` verb when it is applied to an instance of CIM_DNSSettingData. Implementations may support the use of the `set` verb with CIM_DNSSettingData.

The `set` verb is used to modify the properties of the CIM_DNSSettingData instance.

6.2.2.1 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 *DNS Client Profile*.

6.2.2.1.1 Command Form

```
set <CIM_DNSSettingData single instance> <propertyname>=<propertyvalue>
```

6.2.2.1.2 CIM Requirements

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

6.2.2.1.3 Behavior Requirements

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

6.2.2.2 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 *DNS Client Profile*.

6.2.2.2.1 Command Form

```
set <CIM_DNSSettingData single instance> <propertyname1>=<propertyvalue1>
   <propertyname n>=<propertyvalue n>
```

6.2.2.2.2 CIM Requirements

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

6.2.2.2.3 Behavior Requirements

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

6.2.3 Show

This section describes how to implement the `show` verb when applied to an instance of CIM_DNSSettingData. Implementations shall support the use of the `show` verb with CIM_DNSSettingData.

The `show` verb is used to display information about CIM_DNSSettingData.

6.2.3.1 Show a Single Instance

This command form is for the `show` verb applied to a single instance of CIM_DNSSettingData.

6.2.3.1.1 Command Form

```
show <CIM_DNSSettingData single instance>
```

6.2.3.1.2 CIM Requirements

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

6.2.3.1.3 Behavior Requirements

6.2.3.1.3.1 Preconditions

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

6.2.3.1.3.2 Pseudo Code

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

6.2.3.2 Show Multiple Instances

This command form is for the `show` verb applied to multiple instances of CIM_DNSSettingData. This command form corresponds to UFsT-based selection within a scoping system.

6.2.3.2.1 Command Form

```
show <CIM_DNSSettingData multiple instances>
```

6.2.3.2.2 CIM Requirements

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

6.2.3.2.3 Behavior Requirements

6.2.3.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_IPAssignmentSettingData for which related CIM_DNSSettingData instances are displayed. The CIM_DNSSettingData instance is associated with an instance of CIM_IPAssignmentSettingData via an instance of the CIM_OrderedComponent association.

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

6.2.3.2.3.2 Pseudo Code

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

6.3 CIM_DNSProtocolEndpoint

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 instances of 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_DNSProtocolEndpoint

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 mapping is 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_DNSProtocolEndpoint, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering:

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

6.3.2 Set

This section describes how to implement the `set` verb when it is applied to an instance of CIM_DNSProtocolEndpoint. Implementations may support the use of the `set` verb with CIM_DNSProtocolEndpoint.

The `set` verb is used to modify descriptive properties of the CIM_DNSProtocolEndpoint instance.

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 *DNS Client Profile*.

6.3.2.1.1 Command Form

```
set <CIM_DNSProtocolEndpoint single instance> <propertyname>=<propertyvalue>
```

6.3.2.1.2 CIM Requirements

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

6.3.2.1.3 Behavior Requirements

```
$instance=<CIM_DNSProtocolEndpoint single instance>
#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 *DNS Client Profile*.

6.3.2.2.1 Command Form

```
set <CIM_DNSProtocolEndpoint single instance> <propertyname1>=<propertyvalue1>
   <propertynamen>=<propertyvaluen>
```

6.3.2.2.2 CIM Requirements

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

6.3.2.2.3 Behavior Requirements

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

6.3.3 Show

This section describes how to implement the `show` verb when applied to an instance of CIM_DNSProtocolEndpoint. Implementations shall support the use of the `show` verb with CIM_DNSProtocolEndpoint.

The `show` verb is used to display information about the IP interface.

6.3.3.1 Show a Single Instance

This command form is for the `show` verb applied to a single instance of CIM_DNSProtocolEndpoint.

6.3.3.1.1 Command Form

```
show <CIM_DNSProtocolEndpoint single instance>
```

6.3.3.1.2 CIM Requirements

See CIM_DNSProtocolEndpoint in the "CIM Elements" section of the *DNS 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_DNSProtocolEndpoint single instance>
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties };
   }
&smShowInstance ( $instance.getInstancePath(), #propertylist[] );
&smEnd;
```

6.3.3.2 Show Multiple Instances

This command form is for the `show` verb applied to multiple instances of CIM_DNSProtocolEndpoint. This command form corresponds to UFsT-based selection within a scoping system.

6.3.3.2.1 Command Form

```
show <CIM_DNSProtocolEndpoint multiple instances>
```

6.3.3.2.2 CIM Requirements

See CIM_DNSProtocolEndpoint in the "CIM Elements" section of the *DNS 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 we are displaying scoped CIM_DNSProtocolEndpoint instances. The *DNS Client Profile* requires that the CIM_DNSProtocolEndpoint 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_DNSProtocolEndpoint”, “CIM_HostedAccessPoint”,
   $containerInstance.getInstancePath(), #propertylist[] );
&smEnd;
```

6.4 CIM_ElementSettingData

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 instances of 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_ElementSettingData

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	

Command Verb	Requirement	Comments
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_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.

6.4.2 Show

This section describes how to implement the `show` verb when applied to an instance of CIM_ElementSettingData. Implementations shall support the use of the `show` verb with CIM_ElementSettingData.

The `show` command is used to display information about the CIM_ElementSettingData instance or instances.

6.4.2.1 Show Multiple Instances – CIM_DNSProtocolEndpoint Reference

This command form is for the `show` verb applied to multiple instances. This command form corresponds to a `show` command issued against CIM_ElementSettingData where only one reference is specified and the reference is to an instance of CIM_DNSProtocolEndpoint.

6.4.2.1.1 Command Form

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

6.4.2.1.2 CIM Requirements

See CIM_ElementSettingData in the “CIM Elements” section of the *DNS 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_DNSProtocolEndpoint which is referenced by CIM_ElementSettingData.

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

565 #### 6.4.2.1.3.2 Pseudo Code

```
566 #propertylist[] = NULL;
567 if (false == #all)
568    {
569    #propertylist[] = { //all mandatory non-key properties };
570    }
571 &smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getInstancePath(),
572    #propertylist[] );
573 &smEnd;
```

574 ### 6.4.2.2 Show Multiple Instances – CIM_ComputerSystem Reference

575 This command form is for the `show` verb applied to multiple instances. This command form corresponds
576 to a `show` command issued against CIM_ElementSettingData where only one reference is specified and
577 the reference is to an instance of CIM_ComputerSystem.

578 #### 6.4.2.2.1 Command Form

579 **`show <CIM_ElementSettingData `*`multiple instances`*`>`**

580 #### 6.4.2.2.2 CIM Requirements

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

583 #### 6.4.2.2.3 Behavior Requirements

584 ##### 6.4.2.2.3.1 Preconditions

585 `$instance` contains the instance of CIM_ComputerSystem which is referenced by
586 CIM_ElementSettingData.

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

588 ##### 6.4.2.2.3.2 Pseudo Code

```
589 #propertylist[] = NULL;
590 if (false == #all)
591    {
592    #propertylist[] = { //all mandatory non-key properties };
593    }
594 &smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getInstancePath(),
595    #propertylist[] );
596 &smEnd;
```

597 ### 6.4.2.3 Show Multiple Instances – CIM_DNSSettingData Reference

598 This command form is for the `show` verb applied to multiple instances. This command form corresponds
599 to a `show` command issued against CIM_ElementSettingData where only one reference is specified and
600 the reference is to an instance of CIM_DNSSettingData.

601 #### 6.4.2.3.1 Command Form

602 **`show <CIM_ElementSettingData `*`multiple instances`*`>`**

6.4.2.3.2 CIM Requirements

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

6.4.2.3.3 Behavior Requirements

6.4.2.3.3.1 Preconditions

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

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

6.4.2.3.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties };
   }
&smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getInstancePath(),
   #propertylist[] );
&smEnd;
```

6.4.2.4 Show Multiple Instances – CIM_DNSGeneralSettingData Reference

This command form is for the `show` verb applied to multiple instances. This command form corresponds to a `show` command issued against CIM_ElementSettingData where only one reference is specified and the reference is to an instance of CIM_DNSGeneralSettingData.

6.4.2.4.1 Command Form

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

6.4.2.4.2 CIM Requirements

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

6.4.2.4.3 Behavior Requirements

6.4.2.4.3.1 Preconditions

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

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

6.4.2.4.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties };
   }
&smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getInstancePath(),
   #propertylist[] );
&smEnd;
```

6.4.2.5 Show a Single Instance – CIM_DNSGeneralSettingData and CIM_DNSProtocolEndpoint References

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_ElementSettingData where both references are specified and therefore the desired instance is unambiguously identified.

6.4.2.5.1 Command Form

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

6.4.2.5.2 CIM Requirements

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

6.4.2.5.3 Behavior Requirements

6.4.2.5.3.1 Preconditions

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

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

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

6.4.2.5.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties };
   }
&smShowAssociationInstance ( “CIM_ElementSettingData”, $instanceA.getInstancePath(),
   $instanceB.getInstancePath(), #propertylist[] );
&smEnd;
```

6.4.2.6 Show a Single Instance – CIM_DNSGeneralSettingData and CIM_ComputerSystem References

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_ElementSettingData where both references are specified and therefore the desired instance is unambiguously identified.

6.4.2.6.1 Command Form

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

6.4.2.6.2 CIM Requirements

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

6.4.2.6.3 Behavior Requirements

6.4.2.6.3.1 Preconditions

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

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

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

6.4.2.6.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties };
   }
&smShowAssociationInstance ( “CIM_ElementSettingData”, $instanceA.getInstancePath(),
   $instanceB.getInstancePath(), #propertylist[] );
&smEnd;
```

6.4.2.7 Show a Single Instance – CIM_DNSSettingData and CIM_DNSProtocolEndpoint Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_ElementSettingData where both references are specified and therefore the desired instance is unambiguously identified.

6.4.2.7.1 Command Form

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

6.4.2.7.2 CIM Requirements

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

6.4.2.7.3 Behavior Requirements

6.4.2.7.3.1 Preconditions

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

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

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

6.4.2.7.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { //all mandatory non-key properties };
   }
&smShowAssociationInstance ( “CIM_ElementSettingData”, $instanceA.getInstancePath(),
   $instanceB.getInstancePath(), #propertylist[] );
&smEnd;
```

6.5 CIM_SAPSAPDependency

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 instances of 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_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 6.5.2.
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_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.

6.5.2 Show

This section describes how to implement the `show` verb when applied to an instance of CIM_SAPSAPDependency. Implementations shall support the use of the `show` verb with CIM_SAPSAPDependency.

The `show` command is used to display information about the CIM_SAPSAPDependency instance or instances.

6.5.2.1 Show a Single Instance – CIM_IPProtocolEndpoint Reference

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

6.5.2.1.1 Command Form

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

6.5.2.1.2 CIM Requirements

See CIM_SAPSAPDependency in the “CIM Elements” section of the *DNS 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_IPProtocolEndpoint which is referenced by CIM_SAPSAPDependency.

6.5.2.1.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_SAPSAPDependency”, $instance.getInstancePath() );
&smEnd;
```

6.5.2.2 Show a Single Instance – CIM_DNSProtocolEndpoint Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_SAPSAPDependency where the reference specified is to an instance of CIM_DNSProtocolEndpoint. An instance of CIM_DNSProtocolEndpoint is referenced by exactly one instance of CIM_SAPSAPDependency. Therefore, a single instance will be returned.

6.5.2.2.1 Command Form

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

6.5.2.2.2 CIM Requirements

See CIM_SAPSAPDependency in the “CIM Elements” section of the *DNS 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_DNSProtocolEndpoint which is referenced by CIM_SAPSAPDependency.

6.5.2.2.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_SAPSAPDependency”, $instance.getInstancePath() );
&smEnd;
```

6.5.2.3 Show a Single Instance – Both References

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_SAPSAPDependency where both references are specified and therefore the desired instance is unambiguously identified.

6.5.2.3.1 Command Form

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

6.5.2.3.2 CIM Requirements

See CIM_SAPSAPDependency in the "CIM Elements" section of the *DNS 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_IPProtocolEndpoint which is referenced by CIM_SAPSAPDependency.

`$instanceB` contains the instance of CIM_DNSProtocolEndpoint which is referenced by CIM_SAPSAPDependency.

6.5.2.3.3.2 Pseudo Code

```
&smShowAssociationInstance ( "CIM_SAPSAPDependency", $instanceA.getInstancePath(),
   $instanceB.getInstancePath() );
&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 instances of 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

This section describes how to implement the `show` verb when applied to an instance of CIM_HostedAccessPoint. Implementations shall support the use of the `show` verb with CIM_HostedAccessPoint.

The `show` command is used to display information about the CIM_HostedAccessPoint instance or instances.

6.6.2.1 Show Multiple Instances – CIM_ComputerSystem

This command form is for the `show` verb applied to multiple instances. This command form corresponds to a `show` command issued against 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 *DNS 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.getInstancePath() );
&smEnd;
```

6.6.2.2 Show a Single Instance – CIM_DNSProtocolEndpoint Reference

This command form is for the `show` verb applied 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_DNSProtocolEndpoint. An instance of CIM_DNSProtocolEndpoint 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 *DNS 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_DNSProtocolEndpoint which is referenced by CIM_HostedAccessPoint.

6.6.2.2.3.2 Pseudo Code

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

6.6.2.3 Show a Single Instance – CIM_RemoteServiceAccessPoint Reference

This command form is for the `show` verb applied 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 *DNS 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.getInstancePath() );
&smEnd;
```

6.6.2.4 Show a Single Instance – Both References (CIM_DNSProtocolEndpoint)

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_HostedAccessPoint where both references are specified and 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 *DNS 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_DNSProtocolEndpoint which is referenced by CIM_HostedAccessPoint.

6.6.2.4.3.2 Pseudo Code

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

6.6.2.5 Show a Single Instance – Both References (CIM_RemoteServiceAccessPoint)

This command form is for the show verb applied to a single instance. This command form corresponds to a show command issued against CIM_HostedAccessPoint where both references are specified and 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 *DNS 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.getInstancePath(),
   $instanceB.getInstancePath() );
&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 instances of 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.

909 **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	

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

912 ### 6.7.1 Ordering of Results

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

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

917 ### 6.7.2 Show

918 This section describes how to implement the `show` verb when applied to an instance of
919 CIM_RemoteAccessAvailableToElement. Implementations shall support the use of the `show` verb with
920 CIM_RemoteAccessAvailableToElement.

921 The `show` command is used to display information about the CIM_RemoteAccessAvailableToElement
922 instance or instances.

923 #### 6.7.2.1 Show Multiple Instances – CIM_RemoteServiceAccessPoint Reference

924 This command form is for the `show` verb applied to multiple instances. This command form corresponds
925 to a `show` command issued against CIM_RemoteAccessAvailableToElement where only one reference is
926 specified and the reference is to an instance of CIM_RemoteServiceAccessPoint.

927 ##### 6.7.2.1.1 Command Form

928 ```
show <CIM_RemoteAccessAvailableToElement multiple instances>
```

929 ##### 6.7.2.1.2 CIM Requirements

930 See CIM_RemoteAccessAvailableToElement in the “CIM Elements” section of the *DNS Client Profile* for
931 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.

6.7.2.1.3.2 Pseudo Code

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

6.7.2.2 Show Multiple Instances – CIM_DNSProtocolEndpoint Reference

This command form is for the `show` verb applied to multiple instances. This command form corresponds to a `show` command issued against CIM_RemoteAccessAvailableToElement where the reference specified is to an instance of CIM_DNSProtocolEndpoint. The *DNS Client Profile* allows the implementation to model the DNS servers discovered by the client in addition to the DNS Service that actually provides the configuration. Therefore, it is possible for there to be multiple CIM_RemoteAccessAvailableToElement associations that reference the CIM_DNSProtocolEndpoint instance.

6.7.2.2.1 Command Form

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

6.7.2.2.2 CIM Requirements

See CIM_RemoteAccessAvailableToElement in the “CIM Elements” section of the *DNS 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_DNSProtocolEndpoint which is referenced by CIM_RemoteAccessAvailableToElement.

6.7.2.2.3.2 Pseudo Code

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

6.7.2.3 Show a Single Instance – Both References

This command form is for the `show` verb applied to a single instance. This command form corresponds to a `show` command issued against CIM_RemoteAccessAvailableToElement where both references are specified and 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 *DNS Client Profile* for the list of mandatory properties.

970 #### 6.7.2.3.3 Behavior Requirements

971 ##### 6.7.2.3.3.1 Preconditions

972 `$instanceA` contains the instance of CIM_RemoteServiceAccessPoint which is referenced by
973 CIM_RemoteAccessAvailableToElement.

974 `$instanceB` contains the instance of CIM_DNSProtocolEndpoint which is referenced by
975 CIM_RemoteAccessAvailableToElement.

976 ##### 6.7.2.3.3.2 Pseudo Code

```
977 &smShowAssociationInstance ( “CIM_RemoteAccessAvailableToElement”,
978    $instanceA.getInstancePath(), $instanceB.getInstancePath() );
979 &smEnd;
```

980 ## 6.8 CIM_RemoteServiceAccessPoint

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

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

987 **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	Not supported	
show	Shall	See 6.8.2.
start	Not supported	
stop	Not supported	

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

990 ### 6.8.1 Ordering of Results

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

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

6.8.2 Show

This section describes how to implement the `show` verb when applied to an instance of CIM_RemoteServiceAccessPoint. Implementations shall support the use of the `show` verb with CIM_RemoteServiceAccessPoint.

The `show` verb is used to display information about the gateway.

6.8.2.1 Show a Single Instance

This command form is for the `show` verb applied to a single instance of CIM_RemoteServiceAccessPoint.

6.8.2.1.1 Command Form

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

6.8.2.1.2 CIM Requirements

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

6.8.2.1.3 Behavior Requirements

6.8.2.1.3.1 Preconditions

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

6.8.2.1.3.2 Pseudo Code

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

6.8.2.2 Show Multiple Instances

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

6.8.2.2.1 Command Form

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

6.8.2.2.2 CIM Requirements

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

6.8.2.2.3 Behavior Requirements

6.8.2.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ComputerSystem for which scoped CIM_RemoteServiceAccessPoint instances are displayed. The *DNS 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.

6.8.2.2.3.2 Pseudo Code

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

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.0.0	2009-07-14		DMTF Standard Release