Document Number: DSP0816

Date: 2009-07-14

Version: 1.0.0

Host LAN Network Port Profile SM CLP Command Mapping Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Tables

75

Foreword

The *Host LAN Network Port Profile SM CLP Command Mapping Specification* (DSP0816) 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 DTMF Server Management Working Group:

- Aaron Merkin – IBM
- Jon Hass – Dell
- Khachatur Papanyan – 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 *Host LAN Network Port 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 *Host LAN Network Port Profile* using CIM operations.

The target audience for this specification is implementers of the SM CLP support for the *Host LAN Network Port Profile*.

Host LAN Network Port 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 *Host LAN Network Port 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 DSP1035, *Host LAN Network Port Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1035_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).

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

For convenience, Table 1 lists each recipe defined in this mapping which is used for more than one verb or class mapping.

Table 1 – Local Recipes

Recipe Name	Description	Definition
lShowEndpoint	Show an instance of CIM_LANEndpoint	See 5.1.

The following sections detail Local Recipes defined for use in this mapping.

5.1 lShowEndpoint

5.1.1 Description

lShowEndpoint is a reusable recipe for displaying an instance of CIM_LANEndpoint. A recipe is defined for re-use by the `show` and `create` verbs applied to CIM_LANEndpoint.

5.1.2 Preconditions

`$endpoint` contains the instance of CIM_LANEndpoint to display.

`#all` indicates whether the “-all” option was specified.

5.1.3 Pseudo Code

```
sub lShowEndpoint($endpoint, #all)
{
#propertylist[] = NULL;
//if we're not displaying all of the properties, provide a list
if (false == #all)
   {
   #propertylist[] = { “ProtocolIFType”, “MACAddress”, “RequestedState”,
      “EnabledState”, “ElementName” };
   }
&smShowInstance ( $endpoint.GetObjectPath(), #propertyList[] );
&smEnd;
} //lShowEndpoint()
```

6 Mappings

The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in the *Host LAN Network Port 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_ControlledBy

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_ControlledBy

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

Command Verb	Requirement	Comments
reset	Not supported	
set	Not supported	
show	Shall	See 6.2.2.
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_ControlledBy, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering:

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

6.1.2 Show

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

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

6.1.2.1 Show Multiple Instances – CIM_PortController Reference

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

6.1.2.1.1 Command Form

```
show <CIM_ControlledBy multiple instancess>
```

6.1.2.1.2 CIM Requirements

See CIM_ControlledBy in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.1.2.1.3 Behavior Requirements

6.1.2.1.3.1 Preconditions

`$instance` contains the instance of CIM_PortController which is referenced by CIM_ControlledBy.

6.1.2.1.3.2 Pseudo Code

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

6.1.2.2 Show a Single Instance – CIM_NetworkPort 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_ControlledBy where the reference specified is to an instance of CIM_NetworkPort.

6.1.2.2.1 Command Form

```
show <CIM_ControlledBy single instance>
```

6.1.2.2.2 CIM Requirements

See CIM_ControlledBy in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.1.2.2.3 Behavior Requirements

6.1.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_NetworkPort which is referenced by CIM_ControlledBy.

6.1.2.2.3.2 Pseudo Code

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

6.1.2.3 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_ControlledBy where both references are specified and therefore the desired instance is unambiguously identified.

6.1.2.3.1 Command Form

```
show <CIM_ControlledBy single instance>
```

6.1.2.3.2 CIM Requirements

See CIM_ControlledBy in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.1.2.3.3 Behavior Requirements

6.1.2.3.3.1 Preconditions

`$instanceA` contains the instance of CIM_NetworkPort which is referenced by CIM_ControlledBy.

`$instanceB` contains the instance of CIM_PortController which is referenced by CIM_ControlledBy.

6.1.2.3.3.2 Pseudo Code

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

6.2 CIM_ElementCapabilities

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.

298 **Table 3 – 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 6.2.2.
start	Not supported	
stop	Not supported	

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

301 ### 6.2.1 Ordering of Results

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

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

305 ### 6.2.2 Show

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

309 The `show` command is used to display information about the CIM_ElementCapabilities instance or
310 instances.

311 #### 6.2.2.1 Show Multiple Instances – CIM_EnabledLogicalElementCapabilities Reference

312 This command form is for the `show` verb applied to multiple instances. This command form corresponds
313 to a `show` command issued against CIM_ElementCapabilities where only one reference is specified and
314 the reference is to an instance of CIM_EnabledLogicalElementCapabilities.

315 ##### 6.2.2.1.1 Command Form

316 **`show <CIM_ElementCapabilities multiple instances>`**

317 ##### 6.2.2.1.2 CIM Requirements

318 See CIM_ElementCapabilities in the “CIM Elements” section of the *Host LAN Network Port Profile* for the
319 list of mandatory properties.

320 ##### 6.2.2.1.3 Behavior Requirements

321 ###### 6.2.2.1.3.1 Preconditions

322 `$instance` contains the instance of CIM_EnabledLogicalElementCapabilities which is referenced by
323 CIM_ElementCapabilities.

6.2.2.1.3.2 Pseudo Code

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

6.2.2.2 Show a Single Instance – CIM_EnabledLogicalElement 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_ElementCapabilities where the reference specified is to an instance of CIM_EnabledLogicalElement. The *Host LAN Network Port Profile* specifies the usage of CIM_EnabledLogicalElementCapabilities with multiple subclasses of CIM_EnabledLogicalElement. A single instance of CIM_EnabledLogicalElementCapabilities can be associated with each instance of a CIM_EnabledLogicalElement subclass. Therefore, a single instance will be returned.

6.2.2.2.1 Command Form

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

6.2.2.2.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.2.2.2.3 Behavior Requirements

6.2.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_PortController, CIM_NetworkPort, or CIM_LANEndpoint which is referenced by CIM_ElementCapabilities.

6.2.2.2.3.2 Pseudo Code

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

6.2.2.3 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_ElementCapabilities where both references are specified and therefore the desired instance is unambiguously identified.

6.2.2.3.1 Command Form

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

6.2.2.3.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.2.2.3.3 Behavior Requirements

6.2.2.3.3.1 Preconditions

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

360 $instanceB contains the instance of CIM_LANEndpoint, CIM_NetworkPort or CIM_PortController which
361 is referenced by CIM_ElementCapabilities.

362 **6.2.2.3.3.2 Pseudo Code**

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

366 6.3 CIM_EnabledLogicalElementCapabilities

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

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

373 **Table 4 – Command Verb Requirements for CIM_EnabledLogicalElementCapabilities**

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 6.3.2.
start	Not supported	
stop	Not supported	

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

376 6.3.1 Ordering of Results

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

379 • Results for CIM_EnabledLogicalElementCapabilities are unordered; therefore, no algorithm is
380 defined.

381 6.3.2 Show

382 This section describes how to implement the show verb when applied to an instance of
383 CIM_EnabledLogicalElementCapabilities. Implementations shall support the use of the show verb with
384 CIM_EnabledLogicalElementCapabilities.

385 The show verb is used to display information about an instance or instances of the
386 CIM_EnabledLogicalElementCapabilities class.

6.3.2.1 Show a Single Instance

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

6.3.2.1.1 Command Form

```
show <CIM_EnabledLogicalElementCapabilities single instance>
```

6.3.2.1.2 CIM Requirements

See CIM_EnabledLogicalElementCapabilities in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.3.2.1.3 Behavior Requirements

6.3.2.1.3.1 Preconditions

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

6.3.2.1.3.2 Pseudo Code

```
$instance=<CIM_EnabledLogicalElementCapabilities single instance>
#propertylist[] = NULL;
if ( false == #all)
   {
   #propertylist[] = {“RequestedStatesSupported”, “ElementNameEditSupported”,
      “MaxElementNameLen”}
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.3.2.2 Show Multiple Instances

This command form is for the `show` verb applied to multiple instances of CIM_EnabledLogicalElementCapabilities. This command form corresponds to UfsT-based selection within a capabilities collection.

6.3.2.2.1 Command Form

```
show <CIM_EnabledLogicalElementCapabilities multiple instances>
```

6.3.2.2.2 CIM Requirements

See CIM_EnabledLogicalElementCapabilities in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.3.2.2.3 Behavior Requirements

6.3.2.2.3.1 Preconditions

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

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

6.3.2.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all)
   {
   #propertylist[] = {“RequestedStatesSupported”, “ElementNameEditSupported”,
      “MaxElementNameLen”}
   }
&smShowInstances ( “CIM_EnabledLogicalElementCapabilities”, “CIM_MemberOfCollection”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.4 CIM_HostedAccessPoint

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_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 6.4.2.
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`.

6.4.1 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.

6.4.2 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.

6.4.2.1 Show Multiple Instances – CIM_ComputerSystem Reference

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.

6.4.2.1.1 Command Form

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

6.4.2.1.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *Host LAN Network Port 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_ComputerSystem which is referenced by CIM_HostedAccessPoint.

6.4.2.1.3.2 Pseudo Code

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

6.4.2.2 Show a Single Instance – CIM_LANEndpoint 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_LANEndpoint. An instance of CIM_LANEndpoint is referenced by exactly one instance of CIM_HostedAccessPoint. Therefore, a single instance will be returned.

6.4.2.2.1 Command Form

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

6.4.2.2.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.4.2.2.3 Behavior Requirements

6.4.2.2.3.1 Preconditions

$instance contains the instance of CIM_LANEndpoint which is referenced by CIM_HostedAccessPoint.

6.4.2.2.3.2 Pseudo Code

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

6.4.2.3 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_HostedAccessPoint where both references are specified and therefore the desired instance is unambiguously identified.

6.4.2.3.1 Command Form

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

6.4.2.3.2 CIM Requirements

See CIM_HostedAccessPoint in the "CIM Elements" section of the *Host LAN Network Port 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_ComputerSystem which is referenced by CIM_HostedAccessPoint.

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

6.4.2.3.3.2 Pseudo Code

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

6.5 CIM_HostedService

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_HostedService

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

- Results for CIM_HostedService 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_HostedService. Implementations shall support the use of the `show` verb with CIM_HostedService.

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

6.5.2.1 Show Multiple Instances – CIM_ComputerSystem Reference

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

6.5.2.1.1 Command Form

```
show <CIM_HostedService multiple instances>
```

6.5.2.1.2 CIM Requirements

See CIM_HostedService in the “CIM Elements” section of the *Host LAN Network Port 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_ComputerSystem which is referenced by CIM_HostedService.

6.5.2.1.3.2 Pseudo Code

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

6.5.2.2 Show a Single Instance – CIM_NetworkPortConfigurationService 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_HostedService where the reference specified is to an instance of CIM_NetworkPortConfigurationService. An instance of CIM_NetworkPortConfigurationService is referenced by exactly one instance of CIM_HostedService. Therefore, a single instance will be returned.

6.5.2.2.1 Command Form

```
show <CIM_HostedService single instance>
```

6.5.2.2.2 CIM Requirements

See CIM_HostedService in the “CIM Elements” section of the *Host LAN Network Port 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_NetworkPortConfigurationService which is referenced by CIM_HostedService.

6.5.2.2.3.2 Pseudo Code

```
&smShowAssociationInstances ( “CIM_HostedService”, $instance.getObjectPath() );
&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_HostedService where both references are specified and therefore the desired instance is unambiguously identified.

6.5.2.3.1 Command Form

```
show <CIM_HostedService single instance>
```

6.5.2.3.2 CIM Requirements

See CIM_HostedService in the “CIM Elements” section of the *Host LAN Network Port 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_ComputerSystem which is referenced by CIM_HostedService.

`$instanceB` contains the instance of CIM_NetworkPortConfigurationService which is referenced by CIM_HostedService.

6.5.2.3.3.2 Pseudo Code

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

6.6 CIM_LANEndpoint

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.

580 **Table 7 – Command Verb Requirements for CIM_LANEndpoint**

Command Verb	Requirement	Comments
create	May	See 6.6.2.
delete	May	See 6.6.3.
dump	Not supported	
load	Not supported	
reset	May	See 6.6.4.
set	May	See 6.6.5.
show	Shall	See 6.6.6.
start	May	See 6.6.7.
stop	May	See 6.6.8.

581 No mapping is defined for the following verbs for the specified target: `dump` and `load`.

582 ### 6.6.1 Ordering of Results

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

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

586 ### 6.6.2 Create

587 This section describes how to implement the `create` verb when applied to an instance of
588 CIM_LANEndpoint. Implementations may support the use of the `create` verb with CIM_LANEndpoint.

589 The `create` verb is used to create an additional CIM_LANEndpoint instance bound to a
590 CIM_NetworkPort instance.

591 #### 6.6.2.1 Create Specifying Required/Optional Parameters

592 In order to create an instance of CIM_LANEndpoint, a client is required to supply the desired network
593 address. The client can optionally supply one or more group addresses and alias addresses.

594 ##### 6.6.2.1.1 Command Form

```
595 create CIM_LANEndpoint  address=<desiredaddress>
596    groupaddresses=<desiregroupaddresses>]
597    alias addresses=<desiredaliasaddresses>]
598    lanid=<desiredlanid>]
```

599 ##### 6.6.2.1.2 CIM Requirements

```
600 CIM_NetworkPort
601 uint32 CIM_NetworkPortConfigurationService.AddLANEndpoint(
602    [IN]  uint16 Port,
603    [OUT] REF CIM_LANEndpoint Endpoint,
604    [IN] string Address,
605    [IN] string LANID,
606    [IN] string AliasAddresses,
607    [IN] string GroupAddresses);
608 CIM_LANEndpoint
```

6.6.2.1.3 Behavior Requirements

6.6.2.1.3.1 Preconditions

$port contains the CIM_NetworkPort instance for which a new endpoint is created. When using SM ME Addressing, this instance would be the container instance specified in the Resultant Address.

6.6.2.1.3.2 Pseudo Code

```
//the desired address is required, if it is not specified, fail
if (NULL == <desiredaddress>) {
   &smMakeError();
}
#address = <desiredaddress>;
if (NULL != <desiredgroupaddresses>) {
   //convert from command line array assignment to array variable
   #groupaddresses[] = <desiredgroupaddresses>;
}
else  {
   #groupaddresses[] = NULL;
}
if (NULL != <desiredaliasaddresses>) {
   //convert from command line array assignment to array variable
   #aliasaddresses[] = <desiredaliasaddresses>;
}
else  {
   #aliasaddresses[] = NULL;
}
if (NULL != <desiredlanid>)  {
   //convert from command line parameter to variable
   #LANID = <desiredlanid>;
}
else  {
   #LANID = NULL;
}
//find the NetworkPortConfigurationService for the target CIM_NetworkPort
$Services[] = smOpAssociators(
   $instancePath->,
   “CIM_ServiceAffectsElement”,
   “CIM_NetworkPortConfigurationService”,
   NULL,
   NULL);
//there should be one, if not, make an error
if (NULL = $Services[0])  {
   //unsupported
   $OperationError = smNewInstance(“CIM_Error”);
   //CIM_ERR_NOT_SUPPORTED
   $OperationError.CIMStatusCode = 7;
   //Other
   $OperationError.ErrorType = 1;
   //Low
```

```
   $OperationError.PerceivedSeverity = 2;
   $OperationError.OwningEntity = DMTF:SMCLP;
   $OperationError.MessageID = 0x00000001;
   $OperationError.Message = “Operation is not supported.”;
   &smAddError($job, $OperationError);
   &smMakeCommandStatus($job);
   &smEnd;
}
//build the parameter lists and invoke the method
%InArguments[] = {newArgument(“Port”, $Port.GetObjectPath()),
   newArgument (“LANID”, #LANID),
   newArgument (“AliasAddresses”, #aliasaddresses[]),
   newArgument (“GroupAddresses”, #groupaddresses[]),
   %OutArguments[] = { newArgument(“Endpoint”,
                       $Endpoint.GetObjectPath()) };
//invoke method
#returnStatus = smOpInvokeMethod ($Service->,
   “AddLANEndpoint”,
   %InArguments[],
   %OutArguments[]);
// process return code to CLP Command Status
if (0 != #Error.code)  {
   //method invocation failed
   if ( (NULL != #Error.$error) && (NULL != #Error.$error[0]) )    {
      //if the method invocation contains an embedded error
      //use it for the Error for the overall job
      &smAddError($job, #Error.$error[0]);
      &smMakeCommandStatus($job);
      &smEnd;
   }
   else {
      //operation failed, but no detailed error instance, need to make one up
      //make an Error instance and associate with job for Operation
         $OperationError = smNewInstance(“CIM_Error”);
         //CIM_ERR_FAILED
         $OperationError.CIMStatusCode = 1;
         //Software Error
         $OperationError.ErrorType = 4;
         //Unknown
         $OperationError.PerceivedSeverity = 0;
         $OperationError.OwningEntity = DMTF:SMCLP;
         $OperationError.MessageID = 0x00000009;
         $OperationError.Message = “An internal software error has occurred.”;
         &smAddError($job, $OperationError);
         &smMakeCommandStatus($job);
         &smEnd;
   }
}//if CIM op failed
else if (0 == #returnStatus)  {
```

```
   //completed successfully
   &lShowEndpoint($endpoint, “false”);
   &smEnd;
}
else if (4 == #returnStatus)  {
   //generic failure
   $OperationError = smNewInstance(“CIM_Error”);
   //CIM_ERR_FAILED
   $OperationError.CIMStatusCode = 1;
   //Other
   $OperationError.ErrorType = 1;
   //Low
   $OperationError.PerceivedSeverity = 2;
   $OperationError.OwningEntity = DMTF:SMCLP;
   $OperationError.MessageID = 0x00000002;
   $OperationError.Message = “Failed.  No further information is available.”;
   &smAddError($job, $OperationError);
   &smMakeCommandStatus($job);
}
else   {
   //invalid parameter
   $OperationError = smNewInstance(“CIM_Error”);
   //CIM_ERR_FAILED
   $OperationError.CIMStatusCode = 1;
   //Other
   $OperationError.ErrorType = 1;
   //Low
   $OperationError.PerceivedSeverity = 2;
   $OperationError.OwningEntity = DMTF:SMCLP;
   $OperationError.MessageID = 0x00000004;
   $OperationError.Message = “One or more parameters specified are invalid.”;
   &smAddError($job, $OperationError);
   &smMakeCommandStatus($job);
   &smEnd;
}
```

6.6.3 Delete

This section describes how to implement the `delete` verb when applied to an instance of CIM_LANEndpoint. Implementations may support the use of the `delete` verb with CIM_LANEndpoint.

The `delete` command is used to remove an instance of CIM_LANEndpoint which represents a virtual MAC.

6.6.3.1 Delete a Single Instance

Delete a single instance of CIM_LANEndpoint.

6.6.3.1.1 Command Form

```
delete <CIM_LANEndpoint single instance>
```

6.6.3.1.2 CIM Requirements

See CIM_LANEndpoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.6.3.1.3 Behavior Requirements

```
$instance=<CIM_LANEndpoint single instance>
&smOpDeleteInstance ( $instance.GetObjectPath() );
```

6.6.4 Reset

This section describes how to implement the `reset` verb when applied to an instance of CIM_LANEndpoint. Implementations may support the use of the `reset` verb with CIM_LANEndpoint.

The `reset` verb is used to initiate a reset of the CIM_LANEndpoint.

6.6.4.1 Reset a Single Instance

This command form is for the initiation of a reset action against a network endpoint. The mapping is implemented as an invocation of the RequestStateChange( ) method on the instance.

6.6.4.1.1 Command Form

```
reset <CIM_LANEndpoint single instance>
```

6.6.4.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = “reset”,
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.6.4.1.3 Behavior Requirements

```
$instance=<CIM_LANEndpoint single instance>
smResetRSC ( $instance.getObjectPath() );
&smEnd;
```

6.6.5 Set

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

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

6.6.5.1 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 *Host LAN Network Port Profile*.

6.6.5.1.1 Command Form

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

6.6.5.1.2 CIM Requirements

See CIM_LANEndpoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.6.5.1.3 Behavior Requirements

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

6.6.5.2 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 *Host LAN Network Port Profile*.

6.6.5.2.1 Command Form

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

6.6.5.2.2 CIM Requirements

See CIM_LANEndpoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.6.5.2.3 Behavior Requirements

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

6.6.6 Show

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

The `show` verb is used to display information about a network endpoint.

6.6.6.1 Show a Single Instance

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

6.6.6.1.1 Command Form

```
show <CIM_LANEndpoint single instance>
```

6.6.6.1.2 CIM Requirements

See CIM_LANEndpoint in the "CIM Elements" section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.6.6.1.3 Behavior Requirements

6.6.6.1.3.1 Preconditions

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

6.6.6.1.3.2 Pseudo Code

```
$instance=<CIM_LANEndpoint single instance>
&lShowEndpoint ( $instance, #all );
&smEnd;
```

6.6.6.2 Show Multiple Instances Scoped by a System

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

6.6.6.2.1 Command Form

```
show <CIM_LANEndpoint multiple instances>
```

6.6.6.2.2 CIM Requirements

See CIM_LANEndpoint in the "CIM Elements" section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.6.6.2.3 Behavior Requirements

6.6.6.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ComputerSystem for which scoped endpoints (CIM_LANEndpoint instances) are displayed. The *Host LAN Network Port Profile* requires that the CIM_LANEndpoint instance be associated with its scoping system via an instance of the CIM_SystemDevice association.

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

6.6.6.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
//this property list will match the property list in lShowEndpoint()
if (false == #all)
   {
   #propertylist[] = { "ProtocolIFType", "MACAddress", "RequestedState",
      "EnabledState", "ElementName" };
   }
```

```
&smShowInstances ( “CIM_LANEndpoint”, “CIM_SystemDevice”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.6.6.3 Show Multiple Instances Scoped by a Network Port

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

6.6.6.3.1 Command Form

```
show <CIM_LANEndpoint multiple instances>
```

6.6.6.3.2 CIM Requirements

See CIM_LANEndpoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.6.6.3.3 Behavior Requirements

6.6.6.3.3.1 Preconditions

`$containerInstance` contains the instance of CIM_NetworkPort for which scoped endpoints (CIM_LANEndpoint instances) are displayed. The *Host LAN Network Port Profile* requires that the CIM_LANEndpoint instance be associated with a scoping network port via an instance of the CIM_DeviceSAPImplementation association.

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

6.6.6.3.3.2 Pseudo Code

```
#propertylist[] = NULL;
//this property list will match the property list in lShowEndpoint()
if (false == #all)
   {
   #propertylist[] = { “ProtocolIFType”, “MACAddress”, “RequestedState”,
      “EnabledState”, “ElementName” };
   }
&smShowInstances ( “CIM_LANEndpoint”, “CIM_DeviceSAPImplementation”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.6.7 Start

This section describes how to implement the `start` verb when applied to an instance of CIM_LANEndpoint. Implementations may support the use of the `start` verb with CIM_LANEndpoint.

The `start` verb is used to enable an endpoint.

6.6.7.1 Start a Single Instance

This command form is for the `start` verb applied to a single instance of CIM_LANEndpoint.

6.6.7.1.1 Command Form

```
start <CIM_LANEndpoint single instance>
```

6.6.7.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = “enabled”,
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.6.7.1.3 Behavior Requirements

```
$instance=<CIM_LANEndpoint single instance>
smStartRSC ( $instance.getObjectPath() );
&smEnd;
```

6.6.8 Stop

This section describes how to implement the `stop` verb when applied to an instance of CIM_LANEndpoint. Implementations may support the use of the `stop` verb with CIM_LANEndpoint.

The `stop` verb is used to disable an endpoint.

6.6.8.1 Stop a Single Instance

This command form is for the `stop` verb applied to a single instance of CIM_LANEndpoint.

6.6.8.1.1 Command Form

```
stop <CIM_LANEndpoint single instance>
```

6.6.8.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = “disabled”,
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.6.8.1.3 Behavior Requirements

```
$instance=<CIM_LANEndpoint single instance>
smStopRSC ( $instance.getObjectPath() );
&smEnd;
```

6.7 CIM_NetworkPort

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 instances of 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.

934 **Table 8 – Command Verb Requirements for CIM_NetworkPort**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.7.2.
set	May	See 6.7.3.
show	Shall	See 6.7.4.
start	May	See 6.7.5.
stop	May	See 6.7.6.

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

936 ### 6.7.1 Ordering of Results

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

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

940 ### 6.7.2 Reset

941 This section describes how to implement the `reset` verb when applied to an instance of
942 CIM_NetworkPort. Implementations may support the use of the `reset` verb with CIM_NetworkPort.

943 The `reset` verb is used to initiate a reset of the CIM_NetworkPort.

944 #### 6.7.2.1 Reset a Single Instance

945 This command form is for the initiation of a reset action against a single endpoint. The mapping is
946 implemented as an invocation of the RequestStateChange( ) method on the instance.

947 ##### 6.7.2.1.1 Command Form

948 **`reset <CIM_NetworkPort`** ***`single instance`*****`>`**

949 ##### 6.7.2.1.2 CIM Requirements

```
950 uint16 EnabledState;
951 uint16 RequestedState;
952 uint32 EnabledLogicalElement.RequestStateChange (
953    [IN] uint16 RequestedState = “enabled”,
954    [OUT] REF CIM_ConcreteJob Job,
955    [IN] datetime TimeoutPeriod );
```

956 ##### 6.7.2.1.3 Behavior Requirements

```
957 $instance=<CIM_NetworkPort single instance>
958 smResetRSC ( $instance.getObjectPath() );
959 &smEnd;
```

960 ### 6.7.3 Set

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

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

964 #### 6.7.3.1 General Usage of Set for a Single Property

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

967 The requirement for supporting modification of a property using this command form shall be equivalent to
968 the requirement for supporting modification of the property using the ModifyInstance operation as defined
969 in the *Host LAN Network Port Profile*.

970 ##### 6.7.3.1.1 Command Form

971 **`set <CIM_NetworkPort *single instance*> <propertyname>=<propertyvalue>`**

972 ##### 6.7.3.1.2 CIM Requirements

973 See CIM_NetworkPoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of
974 modifiable properties.

975 ##### 6.7.3.1.3 Behavior Requirements

```
976 $instance=<CIM_NetworkPort single instance>
977 #propertyNames[] = {<propertyname>};
978 #propertyValues[] = {<propertyvalue>};
979 &smSetInstance ( $instance, #propertyNames[], #propertyValues[] );
980 &smEnd;
```

981 #### 6.7.3.2 General Usage of Set for Multiple Properties

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

985 The requirement for supporting modification of a property using this command form shall be equivalent to
986 the requirement for supporting modification of the property using the ModifyInstance operation as defined
987 in the *Host LAN Network Port Profile*.

988 ##### 6.7.3.2.1 Command Form

```
989 set <CIM_NetworkPort Single Instance> <propertyname1>=<propertyvalue1>
990    <propertyname n>=<propertyvalue n>
```

991 ##### 6.7.3.2.2 CIM Requirements

992 See CIM_NetworkPoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of
993 supported properties.

994 ##### 6.7.3.2.3 Behavior Requirements

```
995 $instance=<CIM_NetworkPort single instance>
996 #propertyNames[] = {<propertyname>};
997 for #i < n
```

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

6.7.4 Show

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

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

6.7.4.1 Show a Single Instance

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

6.7.4.1.1 Command Form

```
show <CIM_NetworkPort single instance>
```

6.7.4.1.2 CIM Requirements

See CIM_NetworkPoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.7.4.1.3 Behavior Requirements

6.7.4.1.3.1 Preconditions

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

6.7.4.1.3.2 Pseudo Code

```
$instance=<CIM_NetworkPort single instance>
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { “LinkTechnology”, “PermanentAddress”, “DeviceID”,
      “ElementName” };
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.7.4.2 Show Multiple Instances

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

6.7.4.2.1 Command Form

```
show <CIM_NetworkPort multiple instances>
```

6.7.4.2.2 CIM Requirements

See CIM_NetworkPoint in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.7.4.2.3 Behavior Requirements

6.7.4.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ComputerSystem for which scoped network ports (CIM_NetworkPort instances) are displayed. The *Host LAN Network Port Profile* requires that the CIM_NetworkPort instance be associated with its scoping system via an instance of the CIM_SystemDevice association.

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

6.7.4.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { “LinkTechnology”, “PermanentAddress”, “DeviceID”,
      “ElementName” };
   }
&smShowInstances ( “CIM_NetworkPort”, “CIM_SystemDevice”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.7.5 Start

This section describes how to implement the `start` verb when applied to an instance of CIM_NetworkPort. Implementations may support the use of the `start` verb with CIM_NetworkPort.

The `start` verb is used to enable a network port.

6.7.5.1 Start a Single Instance

This command form is for the `start` verb applied to a single instance of CIM_NetworkPort.

6.7.5.1.1 Command Form

```
start <CIM_NetworkPort single instance>
```

6.7.5.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = “enabled”,
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.7.5.1.3 Behavior Requirements

```
$instance=<CIM_NetworkPort single instance>
smStartRSC ( $instance.getObjectPath() );
&smEnd;
```

6.7.6 Stop

This section describes how to implement the `stop` verb when applied to an instance of CIM_NetworkPort. Implementations may support the use of the `stop` verb with CIM_NetworkPort.

The `stop` verb is used to disable a network port.

6.7.6.1 Stop a Single Instance

This command form is for the `stop` verb applied to a single instance of CIM_NetworkPort.

6.7.6.1.1 Command Form

```
stop <CIM_NetworkPort single instance>
```

6.7.6.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = “Disabled”,
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.7.6.1.3 Behavior Requirements

```
$instance=<CIM_NetworkPort single instance>
smStopRSC ( $instance.getObjectPath() );
&smEnd;
```

6.8 CIM_NetworkPortConfigurationService

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 instances of 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_NetworkPortConfigurationService

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	

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

6.8.1 Ordering of Results

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

- Results for CIM_NetworkPortConfigurationService 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_NetworkPortConfigurationService. Implementations shall support the use of the `show` verb with CIM_NetworkPortConfigurationService.

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

6.8.2.1 Show a Single Instance

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

6.8.2.1.1 Command Form

```
show <CIM_NetworkPortConfigurationService single instance>
```

6.8.2.1.2 CIM Requirements

See CIM_NetworkPortConfigurationService in the “CIM Elements” section of the *Host LAN Network Port 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_NetworkPortConfigurationService single instance>
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { “ElementName” };
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.8.2.2 Show Multiple Instances

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

6.8.2.2.1 Command Form

```
show <CIM_NetworkPortConfigurationService multiple instances>
```

6.8.2.2.2 CIM Requirements

See CIM_NetworkPortConfigurationService in the “CIM Elements” section of the *Host LAN Network Port 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 instances of the CIM_NetworkPortConfigurationService are displayed. The *Host LAN Network Port Profile* requires that the CIM_NetworkPortConfigurationService instance be associated with its scoping system via an instance of the CIM_SystemDevice 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[] = { “ElementName” };
   }
&smShowInstances ( “CIM_NetworkPortConfigurationService”, “CIM_SystemDevice”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.9 CIM_PortController

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

Table 10 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 10 is for informational purposes only; in case of a conflict between Table 10 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 10.

Table 10 – Command Verb Requirements for CIM_PortController

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.9.2.
set	May	See 6.9.3.
show	Shall	See 6.9.5.
start	May	See 6.9.6.
stop	May	See 6.9.7.

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

6.9.1 Ordering of Results

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

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

6.9.2 Reset

This section describes how to implement the `reset` verb when applied to an instance of CIM_PortController. Implementations may support the use of the `reset` verb with CIM_PortController.

The `reset` verb is used to initiate a reset of the CIM_PortController.

6.9.2.1 Reset a Single Instance

This command form is for the initiation of a reset action against a port controller. The mapping is implemented as an invocation of the RequestStateChange( ) method on the instance.

6.9.2.1.1 Command Form

```
reset <CIM_PortController single instance>
```

6.9.2.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = “Reset”,
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.9.2.1.3 Behavior Requirements

```
$instance=<CIM_PortController single instance>
smResetRSC ( $instance.getObjectPath() );
&smEnd;
```

6.9.3 Set

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

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

6.9.3.1 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 *Host LAN Network Port Profile*.

6.9.4 Command Form

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

6.9.4.1.1 CIM Requirements

See CIM_PortController in the "CIM Elements" section of the *Host LAN Network Port Profile* for the list of modifiable properties.

6.9.4.1.2 Behavior Requirements

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

6.9.4.2 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 *Host LAN Network Port Profile*.

6.9.4.2.1 Command Form

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

6.9.4.2.2 CIM Requirements

See CIM_PortController in the "CIM Elements" section of the *Host LAN Network Port Profile* for the list of supported properties.

6.9.4.2.3 Behavior Requirements

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

6.9.5 Show

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

The `show` verb is used to display information about the port controller(s).

6.9.5.1 Show a Single Instance

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

6.9.5.1.1 Command Form

```
show <CIM_PortController single instance>
```

6.9.5.1.2 CIM Requirements

See CIM_PortController in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.9.5.1.3 Behavior Requirements

6.9.5.1.3.1 Preconditions

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

6.9.5.1.3.2 Pseudo Code

```
$instance=<CIM_PortController single instance>
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { “ProtocolType”, “MaxUnitsSupported”, “ControllerType”,
      “ElementName” };
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.9.5.2 Show Multiple Instances

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

6.9.5.2.1 Command Form

```
show <CIM_PortController multiple instances>
```

6.9.5.2.2 CIM Requirements

See CIM_PortController in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.9.5.2.3 Behavior Requirements

6.9.5.2.3.1 Preconditions

`$containerInstance` contains the instance of CIM_ComputerSystem for which scoped CIM_PortController instances are displayed. The *Host LAN Network Port Profile* requires that the CIM_PortController instance be associated with its scoping system via an instance of the CIM_SystemDevice association.

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

6.9.5.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)
   {
   #propertylist[] = { “ProtocolType”, “MaxUnitsSupported”, “ControllerType”,
```

```
      "ElementName" };
   }
&smShowInstances ( "CIM_PortController", "CIM_SystemDevice",
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.9.6 Start

This section describes how to implement the `start` verb when applied to an instance of CIM_PortController. Implementations may support the use of the `start` verb with CIM_PortController.

The `start` verb is used to enable a port controller.

6.9.6.1 Start a Single Instance

This command form is for the `start` verb applied to a single instance of CIM_PortController.

6.9.6.1.1 Command Form

```
start <CIM_PortController single instance>
```

6.9.6.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = "Enabled",
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.9.6.1.3 Behavior Requirements

```
$instance=<CIM_PortController single instance>
smStartRSC ( $instance.getObjectPath() );
&smEnd;
```

6.9.7 Stop

This section describes how to implement the `stop` verb when applied to an instance of CIM_PortController. Implementations may support the use of the `stop` verb with CIM_PortController.

The `stop` verb is used to disable a port controller.

6.9.7.1 Stop a Single Instance

This command form is for the `stop` verb applied to a single instance of CIM_PortController.

6.9.7.1.1 Command Form

```
stop <CIM_PortController single instance>
```

6.9.7.1.2 CIM Requirements

```
uint16 EnabledState;
uint16 RequestedState;
uint32 EnabledLogicalElement.RequestStateChange (
   [IN] uint16 RequestedState = "Disabled",
```

```
   [OUT] REF CIM_ConcreteJob Job,
   [IN] datetime TimeoutPeriod );
```

6.9.7.1.3 Behavior Requirements

```
$instance=<CIM_PortController single instance>
smStopRSC ( $instance.getObjectPath() );
&smEnd;
```

6.10 CIM_DeviceSAPImplementation

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

Table 11 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 11 is for informational purposes only; in case of a conflict between Table 11 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 11.

Table 11 – Command Verb Requirements for CIM_DeviceSAPImplementation

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 6.10.2.
start	Not supported	
stop	Not supported	

The following verbs shall not be supported for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.10.1 Ordering of Results

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

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

6.10.2 Show

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

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

6.10.2.1 Show Multiple Instances

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

6.10.2.1.1 Command Form

```
show <CIM_DeviceSAPImplementation multiple instances>
```

6.10.2.1.2 CIM Requirements

See CIM_DeviceSAPImplementation in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.10.2.1.3 Behavior Requirements

6.10.2.1.3.1 Preconditions

`$instance` contains the instance of CIM_PortController which is referenced by CIM_DeviceSAPImplementation.

6.10.2.1.3.2 Pseudo Code

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

6.10.2.2 Show a Single Instance – CIM_NetworkPort 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_DeviceSAPImplementation where the reference specified is to an instance of CIM_NetworkPort. An instance of CIM_NetworkPort is referenced by exactly one instance of CIM_DeviceSAPImplementation. Therefore, a single instance will be returned.

6.10.2.2.1 Command Form

```
show <CIM_DeviceSAPImplementation single instance>
```

6.10.2.2.2 CIM Requirements

See CIM_DeviceSAPImplementation in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.10.2.2.3 Behavior Requirements

6.10.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_NetworkPort which is referenced by CIM_DeviceSAPImplementation.

6.10.2.2.3.2 Pseudo Code

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

6.10.2.3 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_DeviceSAPImplementation where both references are specified and therefore the desired instance is unambiguously identified.

6.10.2.3.1 Command Form

```
show <CIM_DeviceSAPImplementation single instance>
```

6.10.2.3.2 CIM Requirements

See CIM_DeviceSAPImplementation in the "CIM Elements" section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.10.2.3.3 Behavior Requirements

6.10.2.3.3.1 Preconditions

`$instanceA` contains the instance of CIM_PortController which is referenced by CIM_DeviceSAPImplementation.

`$instanceB` contains the instance of CIM_NetworkPort which is referenced by CIM_DeviceSAPImplementation.

6.10.2.3.3.2 Pseudo Code

```
&smShowAssociationInstance ( "CIM_DeviceSAPImplementation",
   $instanceA.getObjectPath(), $instanceB.getObjectPath() );
&smEnd;
```

6.11 CIM_ServiceAffectsElement

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

Table 12 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 12 is for informational purposes only; in case of a conflict between Table 12 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 12.

Table 12 – Command Verb Requirements for CIM_ServiceAffectsElement

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 6.11.2.
start	Not supported	
stop	Not supported	

The following verbs shall not be supported for the specified target: `create`, `delete`, `dump`, `load`, `reset`, `set`, `start`, and `stop`.

6.11.1 Ordering of Results

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

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

6.11.2 Show

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

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

6.11.2.1 Show Multiple Instances

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

6.11.2.1.1 Command Form

`show <CIM_ServiceAffectsElement multiple instances>`

6.11.2.1.2 CIM Requirements

See CIM_ServiceAffectsElement in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.11.2.1.3 Behavior Requirements

6.11.2.1.3.1 Preconditions

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

6.11.2.1.3.2 Pseudo Code

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

6.11.2.2 Show a Single Instance – CIM_NetworkPort

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

6.11.2.2.1 Command Form

`show <CIM_ServiceAffectsElement single instance>`

6.11.2.2.2 CIM Requirements

See CIM_ServiceAffectsElement in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.11.2.2.3 Behavior Requirements

6.11.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_NetworkPort which is referenced by CIM_ServiceAffectsElement.

6.11.2.2.3.2 Pseudo Code

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

6.11.2.3 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_ServiceAffectsElement where both references are specified and therefore the desired instance is unambiguously identified.

6.11.2.3.1 Command Form

```
show <CIM_ServiceAffectsElement single instance>
```

6.11.2.3.2 CIM Requirements

See CIM_ServiceAffectsElement in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.11.2.3.3 Behavior Requirements

6.11.2.3.3.1 Preconditions

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

`$instanceB` contains the instance of CIM_NetworkPort or CIM_PortController which is referenced by CIM_ServiceAffectsElement.

6.11.2.3.3.2 Pseudo Code

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

6.12 CIM_SystemDevice

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

Table 13 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 13 is for informational purposes only; in case of a conflict between Table 13 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 13.

1474 **Table 13 – Command Verb Requirements for CIM_SystemDevice**

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 6.12.2.
start	Not supported	
stop	Not supported	

1475 The following verbs shall not be supported for the specified target: `create`, `delete`, `dump`, `load`,
1476 `reset`, `set`, `start`, and `stop`.

1477 ### 6.12.1 Ordering of Results

1478 When results are returned for multiple instances of CIM_SystemDevice implementations shall utilize the
1479 following algorithm to produce the natural (that is, default) ordering:

1480 - Results for CIM_SystemDevice are unordered; therefore, no algorithm is defined.

1481 ### 6.12.2 Show

1482 This section describes how to implement the `show` verb when applied to an instance of
1483 CIM_SystemDevice. Implementations shall support the use of the `show` verb with CIM_SystemDevice.

1484 The `show` command is used to display information about the CIM_SystemDevice instance or instances.

1485 #### 6.12.2.1 Show Multiple Instances

1486 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1487 to a `show` command issued against CIM_SystemDevice where only one reference is specified and the
1488 reference is to an instance of CIM_ComputerSystem.

1489 ##### 6.12.2.1.1 Command Form

1490 **`show <CIM_SystemDevice multiple instances>`**

1491 ##### 6.12.2.1.2 CIM Requirements

1492 See CIM_SystemDevice in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of
1493 mandatory properties.

1494 ##### 6.12.2.1.3 Behavior Requirements

1495 ###### 6.12.2.1.3.1 Preconditions

1496 `$instance` contains the instance of CIM_ComputerSystem which is referenced by CIM_SystemDevice.

6.12.2.1.3.2 Pseudo Code

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

6.12.2.2 Show a Single Instance – CIM_NetworkPort or CIM_PortController 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_SystemDevice where the reference specified is to an instance of CIM_NetworkPort or CIM_PortController. An instance is referenced by exactly one instance of CIM_SystemDevice. Therefore, a single instance will be returned.

6.12.2.2.1 Command Form

```
show <CIM_SystemDevice single instance>
```

6.12.2.2.2 CIM Requirements

See CIM_SystemDevice in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.12.2.2.3 Behavior Requirements

6.12.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_NetworkPort or CIM_PortController which is referenced by CIM_SystemDevice.

6.12.2.2.3.2 Pseudo Code

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

6.12.2.3 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_SystemDevice where both references are specified and therefore the desired instance is unambiguously identified.

6.12.2.3.1 Command Form

```
show <CIM_SystemDevice single instance>
```

6.12.2.3.2 CIM Requirements

See CIM_SystemDevice in the “CIM Elements” section of the *Host LAN Network Port Profile* for the list of mandatory properties.

6.12.2.3.3 Behavior Requirements

6.12.2.3.3.1 Preconditions

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

`$instanceB` contains the instance of CIM_NetworkPort or CIM_PortController which is referenced by CIM_SystemDevice.

6.12.2.3.3.2 Pseudo Code

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

ANNEX A
(informative)

Change Log

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