Document Number: DSP0838

Date: 2009-06-04

Version: 1.0.0

PCI Device Profile SM CLP Command Mapping Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Tables

Foreword

The *PCI Device Profile SM CLP Command Mapping Specification* (DSP0838) 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

- Ravi Mantena – HP
- Aaron Merkin – IBM
- Brady Evans – HP
- Christina Shaw – HP
- Jon Hass – Dell
- Jeff Hilland – HP
- John Leung – Intel
- Khachatur Papanyan – Dell

Introduction

This document defines the SM CLP mapping for CIM elements described in the *PCI Device Profile*. The information in this specification, combined with the *SM CLP-to-CIM Common Mapping Specification 1.0* (DSP0216), is intended to be sufficient to implement SM CLP commands relevant to the classes, properties, and methods described in the *PCI Device Profile* using CIM operations.

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

PCI Device 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 *PCI Device 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 DSP1075, *PCI Device Profile 1.0.0*, http://www.dmtf.org/standards/published_documents/DSP1075_1.0.0.pdf

DMTF DSP0216, *SM CLP-to-CIM Common Mapping Specification 1.0.0*, http://www.dmtf.org/standards/published_documents/DSP0216_1.0.0.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
PCI
Peripheral Component Interconnect

4.5
PCIe
Peripheral Component Interconnect Express

4.6
SM
Server Management

4.7
SMI-S
Storage Management Initiative Specification

4.8
SNIA
Storage Networking Industry Association

5 Recipes

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

- smShowInstance()
- smShowInstances()
- smSetInstance()
- smShowAssocationInstance()
- smShowAssociationInstances()

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 *PCI Device 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_ElementCapabilities

The `cd`, `exit`, `help`, and `version`, 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_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 6.1.2.
start	Not supported	
stop	Not supported	

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

6.1.1 Ordering of Results

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

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

6.1.2 Show

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

6.1.2.1 Show Command Form for Multiple Instances Target – CIM_EnabledLogicalElementCapabilities Reference

This command form is used to show many instances of CIM_ElementCapabilities. This command form corresponds to a `show` command issued against instances of CIM_ElementCapabilities where only one reference is specified and the reference is to an instance of CIM_EnabledLogicalElementCapabilities.

6.1.2.1.1 Command Form

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

6.1.2.1.2 CIM Requirements

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

6.1.2.1.3 Behavior Requirements

6.1.2.1.3.1 Preconditions

$instance represents the instance of CIM_EnabledLogicalElementCapabilities which is referenced by CIM_ElementCapabilities.

6.1.2.1.3.2 Pseudo Code

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

6.1.2.2 Show Command Form for a Single Instance – CIM_PCIDevice Reference

This command form is used to show a single instance of CIM_ElementCapabilities. This command form corresponds to a `show` command issued against a single instance of CIM_ElementCapabilities where only one reference is specified and the reference is to the instance of CIM_PCIDevice.

6.1.2.2.1 Command Form

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

6.1.2.2.2 CIM Requirements

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

6.1.2.2.3 Behavior Requirements

6.1.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIDevice which is referenced by CIM_ElementCapabilities.

6.1.2.2.3.2 Pseudo Code

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

6.1.2.3 Show Command Form for a Single Instance Target – Both References

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

6.1.2.3.1 Command Form

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

6.1.2.3.2 CIM Requirements

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

6.1.2.3.3 Behavior Requirements

6.1.2.3.3.1 Preconditions

$instanceA represents the referenced instance of CIM_PCIDevice through the CIM_ElementCapabilities association.

$instanceB represents the instance of CIM_EnabledLogicalElementCapabilities which is referenced by CIM_ElementCapabilities.

6.1.2.3.3.2 Pseudo Code

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

6.2 CIM_ControlledBy

The `cd`, `exit`, `help`, and `version` 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	
reset	Not supported	
set	Not supported	
show	Shall	See 6.2.2.
start	Not supported	
stop	Not supported	

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

6.2.1 Ordering of Results

When results are returned for multiple instances of CIM_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.

6.2.2 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.

6.2.2.1 Show Command Form for Multiple Instances Target – CIM_PCIPort Reference

This command form is used to show many instances of CIM_ControlledBy. This command form corresponds to a `show` command issued against instances of CIM_ControlledBy where only one reference is specified and the reference is to an instance of CIM_PCIPort.

6.2.2.1.1 Command Form

```
show <CIM_ControlledBy multiple instances>
```

6.2.2.1.2 CIM Requirements

See CIM_ControlledBy in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.2.2.1.3 Behavior Requirements

6.2.2.1.3.1 Preconditions

$instance represents the instance of CIM_PCIPort which is referenced by CIM_ControlledBy.

6.2.2.1.3.2 Pseudo Code

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

6.2.2.2 Show Command Form for a Single Instance – CIM_PCIDevice Reference

This command form is used to show a single instance of CIM_ControlledBy. This command form corresponds to a `show` command issued against a single instance of CIM_ControlledBy where only one reference is specified and the reference is to the instance of CIM_PCIDevice.

6.2.2.2.1 Command Form

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

6.2.2.2.2 CIM Requirements

See CIM_ControlledBy in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.2.2.2.3 Behavior Requirements

6.2.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIDevice which is referenced by CIM_ControlledBy.

6.2.2.2.3.2 Pseudo Code

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

6.2.2.3 Show Command Form for a Single Instance Target – Both References

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

6.2.2.3.1 Command Form

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

6.2.2.3.2 CIM Requirements

See CIM_ControlledBy in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.2.2.3.3 Behavior Requirements

6.2.2.3.3.1 Preconditions

$instanceA represents the referenced instance of CIM_PCIDevice through the CIM_ControlledBy association.

$instanceB represents the instance of CIM_PCIPort which is referenced by CIM_ControlledBy.

6.2.2.3.3.2 Pseudo Code

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

6.3 CIM_DeviceConnection

The `cd`, `exit`, `help`, and `version` 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_DeviceConnection

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	

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

6.3.1 Ordering of Results

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

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

6.3.2 Show

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

6.3.2.1 Show Command Form for Multiple Instances Target – CIM_PCIPort Reference

This command form is used to show many instances of CIM_DeviceConnection. This command form corresponds to a `show` command issued against instances of CIM_DeviceConnection where only one reference is specified and the reference is to an instance of CIM_PCIPort.

6.3.2.1.1 Command Form

```
show <CIM_DeviceConnection multiple instances>
```

6.3.2.1.2 CIM Requirements

See CIM_DeviceConnection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.3.2.1.3 Behavior Requirements

6.3.2.1.3.1 Preconditions

$instance represents the instance of CIM_PCIPort which is referenced by CIM_DeviceConnection.

6.3.2.1.3.2 Pseudo Code

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

6.3.2.2 Show Command Form for a Single Instance – CIM_PCIPort Reference

This command form is used to show a single instance of CIM_DeviceConnection. This command form corresponds to a show command issued against a single instance of CIM_DeviceConnection where only one reference is specified and the reference is to the instance of CIM_PCIPort.

6.3.2.2.1 Command Form

```
show <CIM_DeviceConnection single instance>
```

6.3.2.2.2 CIM Requirements

See CIM_DeviceConnection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.3.2.2.3 Behavior Requirements

6.3.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIPort which is referenced by CIM_DeviceConnection.

6.3.2.2.3.2 Pseudo Code

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

6.3.2.3 Show Command Form for a Single Instance Target – Both References

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

6.3.2.3.1 Command Form

```
show <CIM_DeviceConnection single instance>
```

6.3.2.3.2 CIM Requirements

See CIM_DeviceConnection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.3.2.3.3 Behavior Requirements

6.3.2.3.3.1 Preconditions

$instanceA represents the referenced instance of CIM_PCIPort through CIM_DeviceConnection association.

$instanceB represents the instance of CIM_PCIPort which is referenced by CIM_DeviceConnection.

6.3.2.3.3.2 Pseudo Code

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

6.4 CIM_HostedCollection

The `cd`, `exit`, `help`, and `version` 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_HostedCollection

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

- Results for CIM_HostedCollection 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_HostedCollection. Implementations shall support the use of the `show` verb with CIM_HostedCollection.

6.4.2.1 Show Command Form for Multiple Instances Target – CIM_ComputerSystem Reference

This command form is used to show many instances of CIM_HostedCollection. This command form corresponds to a `show` command issued against instances of CIM_HostedCollection 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_HostedCollection multiple instances>
```

6.4.2.1.2 CIM Requirements

See CIM_HostedCollection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.4.2.1.3 Behavior Requirements

6.4.2.1.3.1 Preconditions

$instance represents the instance of CIM_ComputerSystem which is referenced by CIM_HostedCollection.

6.4.2.1.3.2 Pseudo Code

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

6.4.2.2 Show Command Form for a Single Instance – CIM_PCIPortGroup Reference

This command form is used to show a single instance of CIM_HostedCollection. This command form corresponds to a `show` command issued against a single instance of CIM_HostedCollection where only one reference is specified and the reference is to the instance of CIM_PCIPortGroup.

6.4.2.2.1 Command Form

```
show <CIM_HostedCollection single instance>
```

6.4.2.2.2 CIM Requirements

See CIM_HostedCollection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.4.2.2.3 Behavior Requirements

6.4.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIPortGroup which is referenced by CIM_HostedCollection.

6.4.2.2.3.2 Pseudo Code

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

6.4.2.3 Show Command Form for a Single Instance Target – Both References

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

6.4.2.3.1 Command Form

```
show <CIM_HostedCollection single instance>
```

6.4.2.3.2 CIM Requirements

See CIM_HostedCollection in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.4.2.3.3 Behavior Requirements

6.4.2.3.3.1 Preconditions

$instanceA represents the referenced instance of CIM_ComputerSystem through the CIM_HostedCollection association.

$instanceB represents the instance of CIM_PCIPortGroup which is referenced by CIM_HostedCollection.

6.4.2.3.3.2 Pseudo Code

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

6.5 CIM_ConcreteIdentity

The `cd`, `exit`, `help`, and `version` 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_ConcreteIdentity

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

- Results for CIM_ConcreteIdentity 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_ConcreteIdentity. Implementations shall support the use of the `show` verb with CIM_ConcreteIdentity.

6.5.2.1 Show Command Form for Multiple Instances Target – CIM_LogicalDevice Reference

This command form is used to show many instances of CIM_ConcreteIdentity. This command form corresponds to a `show` command issued against instances of CIM_ConcreteIdentity where only one reference is specified and the reference is to an instance of CIM_LogicalDevice.

6.5.2.1.1 Command Form

```
show <CIM_ConcreteIdentity multiple instances>
```

6.5.2.1.2 CIM Requirements

See CIM_ConcreteIdentity in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.5.2.1.3 Behavior Requirements

6.5.2.1.3.1 Preconditions

$instance represents the instance of CIM_LogicalDevice which is referenced by CIM_ConcreteIdentity.

6.5.2.1.3.2 Pseudo Code

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

6.5.2.2 Show Command Form for a Single Instance – CIM_PCIDevice Reference

This command form is used to show a single instance of CIM_ConcreteIdentity. This command form corresponds to a `show` command issued against a single instance of CIM_ConcreteIdentity where only one reference is specified and the reference is to the instance of CIM_PCIDevice.

6.5.2.2.1 Command Form

```
show <CIM_ConcreteIdentity single instance>
```

6.5.2.2.2 CIM Requirements

See CIM_ConcreteIdentity in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.5.2.2.3 Behavior Requirements

6.5.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIDevice which is referenced by CIM_ConcreteIdentity.

6.5.2.2.3.2 Pseudo Code

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

6.5.2.3 Show Command Form for a Single Instance Target – Both References

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

6.5.2.3.1 Command Form

```
show <CIM_ConcreteIdentity single instance>
```

6.5.2.3.2 CIM Requirements

See CIM_ConcreteIdentity in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.5.2.3.3 Behavior Requirements

6.5.2.3.3.1 Preconditions

$instanceA represents the referenced instance of CIM_LogicalDevice through CIM_ConcreteIdentity association.

$instanceB represents the instance of CIM_PCIDevice which is referenced by CIM_ConcreteIdentity.

6.5.2.3.3.2 Pseudo Code

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

6.6 CIM_MemberOfCollection

The `cd`, `exit`, `help`, and `version` 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_MemberOfCollection

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

Command Verb	Requirement	Comments
show	Shall	See 6.6.2.
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`.

6.6.1 Ordering of Results

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

- Results for CIM_MemberOfCollection 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_MemberOfCollection. Implementations shall support the use of the `show` verb with CIM_MemberOfCollection.

6.6.2.1 Show Command Form for Multiple Instances Target – CIM_PCIPortGroup Reference

This command form is used to show many instances of CIM_MemberOfCollection. This command form corresponds to a `show` command issued against instances of CIM_MemberOfCollection where only one reference is specified and the reference is to an instance of CIM_PCIPortGroup.

6.6.2.1.1 Command Form

```
show <CIM_MemberOfCollection multiple instances>
```

6.6.2.1.2 CIM Requirements

See CIM_MemberOfCollection in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.6.2.1.3 Behavior Requirements

6.6.2.1.3.1 Preconditions

$instance represents the instance of CIM_PCIPortGroup which is referenced by CIM_MemberOfCollection.

6.6.2.1.3.2 Pseudo Code

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

6.6.2.2 Show Command Form for a Single Instance – CIM_PCIPort Reference

This command form is used to show a single instance of CIM_MemberOfCollection. This command form corresponds to a `show` command issued against a single instance of CIM_MemberOfCollection where only one reference is specified and the reference is to the instance of CIM_PCIPort.

6.6.2.2.1 Command Form

```
show <CIM_MemberOfCollection single instance>
```

6.6.2.2.2 CIM Requirements

See CIM_MemberOfCollection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.6.2.2.3 Behavior Requirements

6.6.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIPort which is referenced by CIM_MemberOfCollection.

6.6.2.2.3.2 Pseudo Code

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

6.6.2.3 Show Command Form for a Single Instance Target – Both References

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

6.6.2.3.1 Command Form

```
show <CIM_MemberOfCollection single instance>
```

6.6.2.3.2 CIM Requirements

See CIM_MemberOfCollection in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.6.2.3.3 Behavior Requirements

6.6.2.3.3.1 Preconditions

$instanceA represents the referenced instance of CIM_PCIPortGroup through the CIM_MemberOfCollection association.

$instanceB represents the instance of CIM_PCIPort which is referenced by CIM_MemberOfCollection.

6.6.2.3.3.2 Pseudo Code

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

6.7 CIM_SystemDevice

The `cd`, `exit`, `help`, and `version` 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.

Table 7 – 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 6.7.2.
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`.

6.7.1 Ordering of Results

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

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

6.7.2 Show

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

6.7.2.1 Show Command Form for Multiple Instances Target – CIM_ComputerSystem Reference

This command form is used to show many instances of CIM_SystemDevice. This command form corresponds to a `show` command issued against instances of CIM_SystemDevice where only one reference is specified and the reference is to an instance of CIM_ComputerSystem.

6.7.2.1.1 Command Form

```
show <CIM_SystemDevice multiple instances>
```

6.7.2.1.2 CIM Requirements

See CIM_SystemDevice in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.7.2.1.3 Behavior Requirements

6.7.2.1.3.1 Preconditions

$instance represents the instance of CIM_ComputerSystem which is referenced by CIM_SystemDevice.

6.7.2.1.3.2 Pseudo Code

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

6.7.2.2 Show Command Form for a Single Instance – CIM_PCIDevice Reference

This command form is used to show a single instance of CIM_SystemDevice. This command form corresponds to a `show` command issued against a single instance of CIM_SystemDevice where only one reference is specified and the reference is to the instance of CIM_PCIDevice.

6.7.2.2.1 Command Form

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

6.7.2.2.2 CIM Requirements

See CIM_SystemDevice in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.7.2.2.3 Behavior Requirements

6.7.2.2.3.1 Preconditions

$instance represents the instance of CIM_PCIDevice which is referenced by CIM_SystemDevice.

6.7.2.2.3.2 Pseudo Code

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

6.7.2.3 Show Command Form for a Single Instance – CIM_PCIPort Reference

This command form is used to show a single instance of CIM_SystemDevice. This command form corresponds to a `show` command issued against a single instance of CIM_SystemDevice where only one reference is specified and the reference is to the instance of CIM_PCIPort.

6.7.2.3.1 Command Form

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

6.7.2.3.2 CIM Requirements

See CIM_SystemDevice in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.7.2.3.3 Behavior Requirements

6.7.2.3.3.1 Preconditions

$instance represents the instance of CIM_PCIPort which is referenced by CIM_SystemDevice.

6.7.2.3.3.2 Pseudo Code

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

6.7.2.4 Show Command Form for a Single Instance Target – CIM_ComputerSystem and CIM_PCIDevice References

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

6.7.2.4.1 Command Form

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

6.7.2.4.2 CIM Requirements

See CIM_SystemDevice in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.7.2.4.3 Behavior Requirements

6.7.2.4.3.1 Preconditions

$instanceA represents the referenced instance of CIM_ComputerSystem through the CIM_SystemDevice association.

$instanceB represents the instance of CIM_PCIDevice which is referenced by CIM_SystemDevice.

6.7.2.4.3.2 Pseudo Code

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

6.7.2.5 Show Command Form for a Single Instance Target – CIM_ComputerSystem and CIM_PCIPort References

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

6.7.2.5.1 Command Form

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

6.7.2.5.2 CIM Requirements

See CIM_SystemDevice in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.7.2.5.3 Behavior Requirements

6.7.2.5.3.1 Preconditions

$instanceA represents the referenced instance of CIM_ComputerSystem through CIM_SystemDevice association.

$instanceB represents the instance of CIM_PCIPort which is referenced by CIM_SystemDevice.

6.7.2.5.3.2 Pseudo Code

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

6.8 CIM_PCIPort

The `cd`, `exit`, `help`, and `version` 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.

Table 8 – Command Verb Requirements for CIM_PCIPort

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`, `load`, `reset`, `set`, `start`, and `stop`.

6.8.1 Ordering of Results

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

- Results for CIM_PCIPort 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_PCIPort. Implementations shall support the use of the `show` verb with CIM_PCIPort.

6.8.2.1 Show Command Form for Multiple Instances Target

This command form is used to show many instances of CIM_PCIPort.

6.8.2.1.1 Command Form

```
show <CIM_PCIPort multiple instances>
```

6.8.2.1.2 CIM Requirements

See CIM_PCIPort in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.8.2.1.3 Behavior Requirements

6.8.2.1.3.1 Preconditions

$containerInstance represents the instance of CIM_ComputerSystem which represents the container system and is associated to the targeted instances of CIM_PCIPort through the CIM_SystemDevice association.

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

6.8.2.1.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstances ( “CIM_PCIPort”, “CIM_SystemDevice”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.8.2.2 Show Command Form for a Single Instance Target

This command form is used to show a single instance of CIM_PCIPort.

6.8.2.2.1 Command Form

`show <CIM_PCIPort` *`single instance`*`>`

6.8.2.2.2 CIM Requirements

See CIM_PCIPort in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.8.2.2.3 Behavior Requirements

6.8.2.2.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIPort.

`$instance=<CIM_PCIPort `*`single instance`*`>;`

#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[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.9 CIM_PCIPortGroup

The `cd`, `exit`, `help`, and `version` 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_PCIPortGroup

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not Support	
show	Shall	See 6.9.2.
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`.

6.9.1 Ordering of Results

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

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

6.9.2 Show

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

6.9.2.1 Show Command Form for Multiple Instances Target

This command form is used to show many instances of CIM_PCIPortGroup.

6.9.2.1.1 Command Form

```
show <CIM_PCIPortGroup multiple instances>
```

6.9.2.1.2 CIM Requirements

See CIM_PCIPortGroup in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.9.2.1.3 Behavior Requirements

6.9.2.1.3.1 Preconditions

$containerInstance represents the instance of CIM_ComputerSystem which represents the container system and is associated to the targeted instances of CIM_PCIPortGroup through the CIM_HostedCollection association.

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

6.9.2.1.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstances ( “CIM_PCIPortGroup”, “CIM_HostedCollection”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.9.2.2 Show Command Form for a Single Instance Target

This command form is used to show a single instance of CIM_PCIPortGroup.

6.9.2.2.1 Command Form

show <CIM_PCIPortGroup *single instance*>

6.9.2.2.2 CIM Requirements

See CIM_PCIPortGroup in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.9.2.2.3 Behavior Requirements

6.9.2.2.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIPortGroup.

```
$instance=<CIM_PCIPortGroup single instance>;
```

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

6.9.2.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.10 CIM_PCIDevice

The `cd`, `exit`, `help`, and `version` 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_PCIDevice

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.10.2.
set	May	See 6.10.3.
show	Shall	See 6.10.4.
start	May	See 6.10.5.
stop	May	See 6.10.6.

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

6.10.1 Ordering of Results

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

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

6.10.2 Reset

6.10.2.1 General Usage of Set for a Single Property

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

6.10.2.1.1 Command Form

```
reset <CIM_PCIDevice single instance>
```

6.10.2.1.2 CIM Requirements

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

6.10.2.1.3 Behavior Requirements

6.10.2.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIDevice.

```
$instance=<CIM_PCIDevice single instance>;
```

6.10.2.1.3.2 Pseudo Code

```
&smResetRSC ( $instance.getObjectPath() );
&smEnd;
```

6.10.3 Set

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

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

6.10.3.1 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 *PCI Device Profile*.

6.10.3.1.1 Command Form

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

6.10.3.1.2 CIM Requirements

See CIM_PCIDevice in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.10.3.1.3 Behavior Requirements

6.10.3.1.3.1 Preconditions

```
$instance=<CIM_PCIDevice single instance>;
```

6.10.3.1.3.2 Pseudo Code

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

6.10.3.2 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 *PCI Device Profile*.

6.10.3.2.1 Command Form

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

6.10.3.2.2 CIM Requirements

See CIM_PCIDevice in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.10.3.2.3 Behavior Requirements

6.10.3.2.3.1 Preconditions

```
$instance=<CIM_PCIDevice single instance>;
```

6.10.3.2.3.2 Pseudo Code

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

6.10.3.3 Set RequestedState to "Enabled"

This section describes how to change the state of the PCI device represented by CIM_PCIDevice to "Enabled".

6.10.3.3.1 Command Form

```
set <CIM_PCIDevice single instance> RequestedState="Enabled"
```

6.10.3.3.2 CIM Requirements

See CIM_PCIDevice in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.10.3.3.3 Behavior Requirements

6.10.3.3.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIDevice.

```
$instance=<CIM_PCIDevice single instance>;
```

6.10.3.3.3.2 Pseudo Code

```
//"Enabled" is valuemap 2
&smRequestStateChange ( $instance.getObjectPath(), 2 );
&smEnd;
```

6.10.4 Show

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

6.10.4.1 Show Command Form for Multiple Instances Target

This command form is used to show many instances of CIM_PCIDevice.

6.10.4.1.1 Command Form

```
show <CIM_PCIDevice multiple instances>
```

6.10.4.1.2 CIM Requirements

See CIM_PCIDevice in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.10.4.1.3 Behavior Requirements

6.10.4.1.3.1 Preconditions

$containerInstance represents the instance of CIM_ComputerSystem which represents the container system and is associated to the targeted instances of CIM_PCIDevice through the CIM_SystemDevice association.

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

6.10.4.1.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstances ( “CIM_PCIDevice”, “CIM_SystemDevice”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.10.4.2 Show Command Form for a Single Instance Target

This command form is used to show a single instance of CIM_PCIDevice.

6.10.4.2.1 Command Form

```
show <CIM_PCIDevice single instance>
```

6.10.4.2.2 CIM Requirements

See CIM_PCIDevice in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.10.4.2.3 Behavior Requirements

6.10.4.2.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIDevice.

```
$instance=<CIM_PCIDevice single instance>;
```

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

6.10.4.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.10.5 Start

6.10.5.1 General Usage of Start for a Single Property

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

6.10.5.1.1 Command Form

```
start <CIM_PCIDevice single instance>
```

6.10.5.1.2 CIM Requirements

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

6.10.5.1.3 Behavior Requirements

6.10.5.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIDevice.

```
$instance=<CIM_PCIDevice single instance>;
```

6.10.5.1.3.2 Pseudo Code

```
&smStartRSC ( $instance.getObjectPath() );
&smEnd;
```

6.10.6 Stop

6.10.6.1 General Usage of Stop for a Single Property

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

6.10.6.1.1 Command Form

```
stop <CIM_PCIDevice single instance>
```

6.10.6.1.2 CIM Requirements

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

6.10.6.1.3 Behavior Requirements

6.10.6.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIDevice.

```
$instance=<CIM_PCIDevice single instance>;
```

6.10.6.1.3.2 Pseudo Code

```
&smStopRSC ( $instance.getObjectPath() );
&smEnd;
```

6.11 CIM_PCIBridge

The `cd`, `exit`, `help`, and `version` 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_PCIBridge

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.11.2.
set	May	See 6.11.3.
show	Shall	See 6.11.4.
start	May	See 6.11.5.
stop	May	See 6.11.6.

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

6.11.1 Ordering of Results

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

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

6.11.2 Reset

6.11.2.1 General Usage of Reset for a Single Property

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

6.11.2.1.1 Command Form

```
reset <CIM_PCIBridge single instance>
```

6.11.2.1.2 CIM Requirements

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

6.11.2.1.3 Behavior Requirements

6.11.2.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIBridge.

```
$instance=<CIM_PCIBridge single instance>;
```

6.11.2.1.3.2 Pseudo Code

```
&smResetRSC ( $instance.getObjectPath() );
&smEnd;
```

6.11.3 Set

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

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

6.11.3.1 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 *PCI Device Profile*.

6.11.3.1.1 Command Form

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

6.11.3.1.2 CIM Requirements

See CIM_PCIBridge in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.11.3.1.3 Behavior Requirements

6.11.3.1.3.1 Preconditions

```
$instance=<CIM_PCIBridge single instance>
```

6.11.3.1.3.2 Pseudo Code

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

6.11.3.2 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 *PCI Device Profile*.

6.11.3.2.1 Command Form

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

6.11.3.2.2 CIM Requirements

See CIM_PCIBridge in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.11.3.2.3 Behavior Requirements

6.11.3.2.3.1 Preconditions

```
$instance=<CIM_PCIBridge single instance>;
```

6.11.3.2.3.2 Pseudo Code

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

6.11.3.3 Set RequestedState to “Enabled”

This section describes how to change the state of the PCI device represented by CIM_PCIBridge to “Enabled”.

6.11.3.3.1 Command Form

```
set <CIM_PCIBridge single instance> RequestedState=”Enabled”
```

6.11.3.3.2 CIM Requirements

See CIM_PCIBridge in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.11.3.3.3 Behavior Requirements

6.11.3.3.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIBridge.

```
$instance=<CIM_PCIBridge single instance>;
```

6.11.3.3.3.2 Pseudo Code

```
//"Enabled" is valuemap 2
&smRequestStateChange ( $instance.getObjectPath(), 2 );
&smEnd;
```

6.11.4 Show

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

6.11.4.1 Show Command Form for Multiple Instances Target

This command form is used to show many instances of CIM_PCIBridge.

6.11.4.1.1 Command Form

```
show <CIM_PCIBridge multiple instances>
```

6.11.4.1.2 CIM Requirements

See CIM_PCIBridge in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.11.4.1.3 Behavior Requirements

6.11.4.1.3.1 Preconditions

$containerInstance represents the instance of CIM_ComputerSystem which represents the container system and is associated to the targeted instances of CIM_PCIBridge through the CIM_SystemDevice association.

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

6.11.4.1.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstances ( "CIM_PCIBridge", "CIM_SystemDevice",
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.11.4.2 Show Command Form for a Single Instance Target

This command form is used to show a single instance of CIM_PCIBridge.

6.11.4.2.1 Command Form

```
show <CIM_PCIBridge single instance>
```

6.11.4.2.2 CIM Requirements

See CIM_PCIBridge in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.11.4.2.3 Behavior Requirements

6.11.4.2.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIBridge.

```
$instance=<CIM_PCIBridge single instance>;
```

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

6.11.4.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all )
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.11.5 Start

6.11.5.1 General Usage of Start for a Single Property

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

6.11.5.1.1 Command Form

```
start <CIM_PCIBridge single instance>
```

6.11.5.1.2 CIM Requirements

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

6.11.5.1.3 Behavior Requirements

6.11.5.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIBridge.

```
$instance=<CIM_PCIBridge single instance>;
```

6.11.5.1.3.2 Pseudo Code

```
&smStartRSC ( $instance.getObjectPath() );
&smEnd;
```

6.11.6 Stop

6.11.6.1 General Usage of Stop for a Single Property

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

6.11.6.1.1 Command Form

```
stop <CIM_PCIBridge single instance>
```

6.11.6.1.2 CIM Requirements

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

6.11.6.1.3 Behavior Requirements

6.11.6.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIBridge.

```
$instance=<CIM_PCIBridge single instance>;
```

6.11.6.1.3.2 Pseudo Code

```
&smStopRSC ( $instance.getObjectPath() );
&smEnd;
```

6.12 CIM_PCIeSwitch

The `cd`, `exit`, `help`, and `version` 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_PCIeSwitch

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

Command Verb	Requirement	Comments
reset	May	See 6.12.2.
set	May	See 6.12.3.
show	Shall	See 6.12.4.
start	May	See 6.12.5.
stop	May	See 6.12.6.

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

6.12.1 Ordering of Results

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

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

6.12.2 Reset

6.12.2.1 General Usage of Reset for a Single Property

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

6.12.2.1.1 Command Form

```
reset <CIM_PCIeSwitch single instance>
```

6.12.2.1.2 CIM Requirements

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

6.12.2.1.3 Behavior Requirements

6.12.2.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIeSwitch.

```
$instance=<CIM_PCIeSwitch single instance>;
```

6.12.2.1.3.2 Pseudo Code

```
&smResetRSC ( $instance.getObjectPath() );
&smEnd;
```

6.12.3 Set

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

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

6.12.3.1 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 *PCI Device Profile*.

6.12.3.1.1 Command Form

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

6.12.3.1.2 CIM Requirements

See CIM_PCIeSwitch in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.12.3.1.3 Behavior Requirements

6.12.3.1.3.1 Preconditions

```
$instance=<CIM_PCIeSwitch single instance>;
```

6.12.3.1.3.2 Pseudo Code

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

6.12.3.2 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 *PCI Device Profile*.

6.12.3.2.1 Command Form

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

6.12.3.2.2 CIM Requirements

See CIM_PCIeSwitch in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.12.3.2.3 Behavior Requirements

6.12.3.2.3.1 Preconditions

```
$instance=<CIM_PCIeSwitch single instance>;
```

6.12.3.2.3.2 Pseudo Code

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

6.12.3.3 Set RequestedState to "Enabled"

This section describes how to change the state of the PCI device represented by CIM_PCIeSwitch to "Enabled".

6.12.3.3.1 Command Form

```
set <CIM_PCIeSwitch single instance> RequestedState="Enabled"
```

6.12.3.3.2 CIM Requirements

See CIM_PCIeSwitch in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.12.3.3.3 Behavior Requirements

6.12.3.3.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIeSwitch.

```
$instance=<CIM_PCIeSwitch single instance>;
```

6.12.3.3.3.2 Pseudo Code

```
//"Enabled" is valuemap 2
&smRequestStateChange ( $instance.getObjectPath(), 2 );
&smEnd;
```

6.12.4 Show

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

6.12.4.1 Show Command Form for Multiple Instances Target

This command form is used to show many instances of CIM_PCIeSwitch.

6.12.4.1.1 Command Form

```
show <CIM_PCIeSwitch multiple instances>
```

6.12.4.1.2 CIM Requirements

See CIM_PCIeSwitch in the "CIM Elements" section of the *PCI Device Profile* for the list of mandatory properties.

6.12.4.1.3 Behavior Requirements

6.12.4.1.3.1 Preconditions

$containerInstance represents the instance of CIM_ComputerSystem which represents the container system and is associated to the targeted instances of CIM_PCIeSwitch through the CIM_SystemDevice association.

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

6.12.4.1.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all)
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstances ( “CIM_PCIeSwitch”, “CIM_SystemDevice”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.12.4.2 Show Command Form for a Single Instance Target

This command form is used to show a single instance of CIM_PCIeSwitch.

6.12.4.2.1 Command Form

show <CIM_PCIeSwitch *single instance*>

6.12.4.2.2 CIM Requirements

See CIM_PCIeSwitch in the “CIM Elements” section of the *PCI Device Profile* for the list of mandatory properties.

6.12.4.2.3 Behavior Requirements

6.12.4.2.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIeSwitch.

```
$instance=<CIM_PCIeSwitch single instance>;
```

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

6.12.4.2.3.2 Pseudo Code

```
#propertylist[] = NULL;
if ( false == #all)
   {
   #propertylist[] = <array of mandatory non-key property names (see CIM
      Requirements)>;
   }
&smShowInstance( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.12.5 Start

6.12.5.1 General Usage of Start for a Single Property

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

6.12.5.1.1 Command Form

```
start <CIM_PCIeSwitch single instance>
```

6.12.5.1.2 CIM Requirements

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

6.12.5.1.3 Behavior Requirements

6.12.5.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIeSwitch.

```
$instance=<CIM_PCIeSwitch single instance>;
```

6.12.5.1.3.2 Pseudo Code

```
&smStartRSC ( $instance.getObjectPath() );
&smEnd;
```

6.12.6 Stop

6.12.6.1 General Usage of Stop for a Single Property

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

6.12.6.1.1 Command Form

```
stop <CIM_PCIeSwitch single instance>
```

6.12.6.1.2 CIM Requirements

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

6.12.6.1.3 Behavior Requirements

6.12.6.1.3.1 Preconditions

$instance represents the targeted instance of CIM_PCIeSwitch.

```
$instance=<CIM_PCIeSwitch single instance>;
```

6.12.6.1.3.2 Pseudo Code

```
&smStopRSC ( $instance.getObjectPath() );
&smEnd;
```

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.0.0	2009-06-04		DMTF Standard Release