Document Number: DSP0804

Date: 2009-06-04

Version: 1.0.0

Modular Systems Profile SM CLP Command Mapping Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party’s reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

CONTENTS

Tables

I

Foreword

The *Modular Systems Profile SM CLP Command Mapping Specification* (DSP0804) 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 *Modular Systems 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 *Modular Systems Profile* using CIM operations.

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

Modular Systems 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 *Modular Systems 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 DSP1008, *Modular Systems Profile 1.0*, http://www.dmtf.org/standards/published_documents/DSP1008_1.0.pdf

DMTF DSP0216, *SM CLP-to-CIM Common Mapping Specification 1.0*, http://www.dmtf.org/standards/published_documents/DSP0216_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

IETF, RFC 2119, *Key words for Use in RFCs to Indicate Requirement Levels*, March 1997, http://www.ietf.org/rfc/rfc2119.txt

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

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
- smShowAssociationInstance
- smShowAssociationInstances

6 Mappings

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

6.1 CIM_AdminDomain

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

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

Table 1 – Command Verb Requirements for CIM_AdminDomain

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

Command Verb	Requirement	Comments
set	Not supported	
show	Shall	See 6.1.2.
start	Not supported	
stop	Not supported	

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

6.1.1 Ordering of Results

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

- Results for CIM_AdminDomain 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_AdminDomain. Implementations shall support the use of the `show` verb with CIM_AdminDomain.

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

6.1.2.1 Show a Single Instance

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

6.1.2.1.1 Command Form

```
show <CIM_AdminDomain single instance>
```

6.1.2.1.2 CIM Requirements

See CIM_AdminDomain in the “CIM_AdminDomain – Power Domain” and "CIM_AdminDomain – Cooling Domain" sections of the *Modular Systems Profile* for the list of mandatory properties.

6.1.2.1.3 Behavior Requirements

6.1.2.1.3.1 Preconditions

In this section, `$instance` represents the targeted instance of CIM_AdminDomain.

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

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

6.1.2.1.3.2 Pseudo Code

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

6.1.2.2 Show Multiple Instances

This command form is for the `show` verb applied to multiple instances of CIM_AdminDomain. Each instance of CIM_AdminDomain is scoped to a CIM_ComputerSystem instance through the CIM_SystemComponent association.

6.1.2.2.1 Command Form

```
show <CIM_AdminDomain multiple instances>
```

6.1.2.2.2 CIM Requirements

See CIM_AdminDomain in the “CIM_AdminDomain – Power Domain” and "CIM_AdminDomain – Cooling Domain" sections of the *Modular Systems Profile* for the list of mandatory properties.

6.1.2.2.3 Behavior Requirements

6.1.2.2.3.1 Preconditions

In this section, `$containerInstance` represents the instance of CIM_ComputerSystem which represents the container system and is associated to the targeted instances of CIM_AdminDomain through the CIM_SystemComponent association.

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

6.1.2.2.3.2 Pseudo Code

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

6.2 CIM_ComputerSystem

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 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_ComputerSystem

Command Verb	Requirement	Comments
Create	Not supported	
Delete	Not supported	
Dump	Not supported	
Load	Not supported	
Reset	Not supported	

Command Verb	Requirement	Comments
Set	Not supported	
Show	Shall	See 6.2.2.
Start	Not supported	
Stop	Not supported	

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

6.2.1 Ordering of Results

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

- Results for CIM_ComputerSystem 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_ComputerSystem. Implementations shall support the use of the `show` verb with CIM_ComputerSystem.

6.2.2.1 Show Command Form for Multiple Instances Target

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

6.2.2.1.1 Command Form

```
show <CIM_ComputerSystem multiple instances>
```

6.2.2.1.2 CIM Requirements

See CIM_ComputerSystem in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.2.2.1.3 Behavior Requirements

6.2.2.1.3.1 Preconditions

In this section, `$containerInstance` represents the instance of a subclass of CIM_System which represents the container system and is associated to the targeted instances of CIM_ComputerSystem through the CIM_SystemComponent association.

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

6.2.2.1.3.2 Pseudo Code

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

6.2.2.2 Show Command Form for a Single Instance Target

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

6.2.2.2.1 Command Form

```
show <CIM_ComputerSystem single instance>
```

6.2.2.2.1.1 CIM Requirements

See CIM_ComputerSystem in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.2.2.2.2 Behavior Requirements

6.2.2.2.2.1 Preconditions

In this section, `$instance` represents the targeted instance of CIM_ComputerSystem.

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

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

6.2.2.2.2.2 Pseudo Code

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

6.3 CIM_ConcreteDependency

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

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

Table 3 – Command Verb Requirements for CIM_ConcreteDependency

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.

Command Verb	Requirement	Comments
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_ConcreteDependency, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering:

- Results for CIM_ConcreteDependency 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_ConcreteDependency. Implementations shall support the use of the `show` verb with CIM_ConcreteDependency.

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

6.3.2.1 Show Multiple Instances – CIM_PhysicalPackage Reference

This command form is for the `show` verb applied to multiple instances. This command form corresponds to a `show` command issued against CIM_ConcreteDependency where only one reference is specified and the reference is to an instance of CIM_PhysicalPackage representing a blade package.

6.3.2.1.1 Command Form

```
show <CIM_ConcreteDependency multiple instances>
```

6.3.2.1.2 CIM Requirements

See CIM_ConcreteDependency in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.3.2.1.3 Behavior Requirements

6.3.2.1.3.1 Preconditions

`$instance` contains the instance of CIM_PhysicalPackage which is referenced by CIM_ConcreteDependency.

6.3.2.1.3.2 Pseudo Code

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

6.3.2.2 Show a Single Instance – CIM_PhysicalPackage 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_ConcreteDependency where the reference specified is to an instance of CIM_PhysicalPackage representing a blade expansion. An instance of CIM_PhysicalPackage representing a blade expansion is referenced by exactly one instance of CIM_ConcreteDependency. Therefore, a single instance will be returned.

6.3.2.2.1 Command Form

```
show <CIM_ConcreteDependency single instance>
```

6.3.2.2.2 CIM Requirements

See CIM_ConcreteDependency in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.3.2.2.3 Behavior Requirements

6.3.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_PhysicalPackage representing a blade expansion that is referenced by CIM_ConcreteDependency.

6.3.2.2.3.2 Pseudo Code

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

6.3.2.3 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_ConcreteDependency where both references are specified and therefore the desired instance is unambiguously identified.

6.3.2.3.1 Command Form

```
show <CIM_ConcreteDependency single instance>
```

6.3.2.3.2 CIM Requirements

See CIM_ConcreteDependency in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.3.2.3.3 Behavior Requirements

6.3.2.3.3.1 Preconditions

`$instanceA` contains the instance of CIM_PhysicalPackage representing a blade that is referenced by CIM_ConcreteDependency.

`$instanceB` contains the instance of CIM_PhysicalPackage representing a blade expansion that is referenced by CIM_ConcreteDependency.

6.3.2.3.3.2 Pseudo Code

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

6.4 CIM_SystemComponent

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 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_SystemComponent

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

- Results for CIM_SystemComponent 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_SystemComponent. Implementations shall support the use of the `show` verb with CIM_SystemComponent.

6.4.2.1 Show Command Form for Multiple Instances – Modular System Reference

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

6.4.2.1.1 Command Form

```
show <CIM_SystemComponent multiple instances>
```

6.4.2.1.2 CIM Requirements

See CIM_SystemComponent in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.4.2.1.3 Behavior Requirements

6.4.2.1.3.1 Preconditions

In this section, `$instance` represents the instance of a CIM_ComputerSystem which is referenced by CIM_SystemComponent.

6.4.2.1.3.2 Pseudo Code

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

6.4.2.2 Show Command Form for Multiple Instances – Chassis Manager Reference

This command form is used to show a single instance of CIM_SystemComponent. This command form corresponds to a `show` command issued against a single instance of CIM_SystemComponent, where only one reference is specified and the reference is to the instance of CIM_ComputerSystem that represents a Chassis Manager. CIM_SystemComponent will associate the CIM_ComputerSystem instance for the chassis manager to the modular system and could associate it to one or more CIM_AdminDomain instances.

6.4.2.2.1 Command Form

```
show <CIM_SystemComponent multiple instances>
```

6.4.2.2.2 CIM Requirements

See CIM_SystemComponent in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.4.2.2.3 Behavior Requirements

6.4.2.2.3.1 Preconditions

In this section `$instance` represents the instance of CIM_ComputerSystem which is referenced by CIM_SystemComponent.

6.4.2.2.3.2 Pseudo Code

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

6.4.2.3 Show Command Form for Multiple Instances– Processor Blade Reference

This command form is used to show a single instance of CIM_SystemComponent. This command form corresponds to a `show` command issued against a single instance of CIM_SystemComponent, where only one reference is specified and the reference is to the instance of CIM_ComputerSystem that represents a Processor Blade. CIM_SystemComponent will associate the CIM_ComputerSystem instance for the processor blade to the modular system and could associate it to one or more CIM_AdminDomain instances.

6.4.2.3.1 Command Form

```
show <CIM_SystemComponent multiple instances>
```

6.4.2.3.2 CIM Requirements

See CIM_SystemComponent in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.4.2.3.3 Behavior Requirements

6.4.2.3.3.1 Preconditions

In this section `$instance` represents the instance of CIM_ComputerSystem which is referenced by CIM_SystemComponent.

6.4.2.3.3.2 Pseudo Code

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

6.4.2.4 Show Command Form for Multiple Instances – Cooling Domain Reference

This command form is used to show a single instance of CIM_SystemComponent. This command form corresponds to a `show` command issued against a single instance of CIM_SystemComponent, where only one reference is specified and the reference is to the instance of CIM_AdminDomain that represents a cooling domain. CIM_SystemComponent will associate the CIM_AdminDomain instance for the cooling domain to the modular system and could associate it to one or more contained CIM_ManagedSystemElement instances.

6.4.2.4.1 Command Form

```
show <CIM_SystemComponent single instance>
```

6.4.2.4.2 CIM Requirements

See CIM_SystemComponent in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.4.2.4.3 Behavior Requirements

6.4.2.4.3.1 Preconditions

In this section, `$instance` represents the instance of CIM_AdminDomain which is referenced by CIM_SystemComponent.

6.4.2.4.3.2 Pseudo Code

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

6.4.2.5 Show Command Form for Multiple Instances – Power Domain Reference

This command form is used to show a single instance of CIM_SystemComponent. This command form corresponds to a `show` command issued against a single instance of CIM_SystemComponent, where only one reference is specified and the reference is to the instance of CIM_AdminDomain that represents a power domain. CIM_SystemComponent will associate the CIM_AdminDomain instance for the power domain to the modular system and could associate it to one or more contained CIM_ManagedSystemElement instances.

6.4.2.5.1 Command Form

```
show <CIM_SystemComponent single instance>
```

6.4.2.5.2 CIM Requirements

See CIM_SystemComponent in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.4.2.5.3 Behavior Requirements

6.4.2.5.3.1 Preconditions

In this section, `$instance` represents the instance of CIM_AdminDomain which is referenced by CIM_SystemComponent.

6.4.2.5.3.2 Pseudo Code

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

6.4.2.6 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 a `show` command issued against CIM_SystemComponent where both references are specified and therefore the desired instance is unambiguously identified.

6.4.2.6.1 Command Form

```
show <CIM_SystemComponent single instance>
```

6.4.2.6.2 CIM Requirements

See CIM_SystemComponent in the “CIM Elements” section of the *Modular Systems Profile* for the list of mandatory properties.

6.4.2.6.3 Behavior Requirements

6.4.2.6.3.1 Preconditions

In this section, `$instanceA` represents the instance of CIM_ManagedSystemElement referenced by the CIM_SystemComponent association. `$instanceB` represents the instance of a subclass of CIM_System that is referenced by CIM_SystemComponent.

6.4.2.6.3.2 Pseudo Code

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

ANNEX A
(informative)

Change Log

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