Document Number: DSP0831

Date: 2009-06-04

Version: 1.0.0

Platform Watchdog Profile SM CLP 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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

Foreword

The *Platform Watchdog Profile SM CLP Mapping Specification* (DSP0831) was prepared by the Server Management Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Conventions

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

Acknowledgements

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

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

Introduction

This document defines the SM CLP mapping for CIM elements described in *Platform Watchdog Profile*. The information in this specification, combined with *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 *Platform Watchdog Profile* using CIM operations.

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

Platform Watchdog Profile SM CLP 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 *Platform Watchdog 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 DSP1040, *Platform Watchdog Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1040_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

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

3 Terms and Definitions

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

3.1
can
used for statements of possibility and capability, whether material, physical, or causal

3.2
cannot
used for statements of possibility and capability, whether material, physical, or causal

3.3
conditional
indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met

3.4
mandatory
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.5

may

indicates a course of action permissible within the limits of the document

3.6

need not

indicates a course of action permissible within the limits of the document

3.7

optional

indicates a course of action permissible within the limits of the document

3.8

shall

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

3.9

shall not

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

3.10

should

indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.11

should not

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

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1

CIM

Common Information Model

4.2

CLP

Command Line Protocol

4.3

DMTF

Distributed Management Task Force

4.4

IETF

Internet Engineering Task Force

4.5

SM

Server Management

4.6

SMI-S

Storage Management Initiative Specification

4.7

SNIA

Storage Networking Industry Association

4.8

UFsT

User Friendly selection Tag

5 Recipes

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

- smShowInstance()
- smShowInstances()
- smSetInstance()
- smShowAssociationInstance()
- smShowAssociationInstances()

6 Mappings

The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in the *Platform Watchdog 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_DeviceServiceImplementation

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 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_DeviceServiceImplementation

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

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

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

6.1.1 Ordering of Results

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

- Results for CIM_DeviceServiceImplementation 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_DeviceServiceImplementation. Implementations shall support the use of the `show` verb with CIM_DeviceServiceImplementation.

6.1.2.1 Show Command Form for Multiple Instances – CIM_PlatformWatchdogService Reference

This command form is used to show many instances of CIM_DeviceServiceImplementation. This command form corresponds to the `show` command issued against instances of CIM_DeviceServiceImplementation where only one reference is specified and the reference is to the instance of CIM_PlatformWatchdogService.

6.1.2.1.1 Command Form

```
show <CIM_DeviceServiceImplementation multiple instances>
```

6.1.2.1.2 CIM Requirements

See CIM_DeviceServiceImplementation in the “CIM Elements” section of the *Platform Watchdog 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_PlatformWatchdogService which is referenced by CIM_DeviceServiceImplementation.

6.1.2.1.3.2 Pseudo Code

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

6.1.2.2 Show Command Form for Multiple Instances – CIM_LogicalDevice Reference

This command form is used to show many instances of CIM_DeviceServiceImplementation. The command form corresponds to the show command issued against the instance of CIM_DeviceServiceImplementation where only one reference is specified and the reference is to the instance of a concrete subclass of CIM_LogicalDevice.

6.1.2.2.1 Command Form

```
show <CIM_DeviceServiceImplementation multiple instances>
```

6.1.2.2.2 CIM Requirements

See CIM_DeviceServiceImplementation in the “CIM Elements” section of the *Platform Watchdog 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 a concrete subclass of CIM_LogicalDevice which is referenced by CIM_DeviceServiceImplementation.

6.1.2.2.3.2 Pseudo Code

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

6.1.2.3 Show Command Form for a Single Instance – Both References

This command form is used to show a single instance of CIM_DeviceServiceImplementation. This command form corresponds to the show command issued against instances of CIM_DeviceServiceImplementation where both references are specified and therefore the desired instance is unambiguously identified.

6.1.2.3.1 Command Form

```
show <CIM_DeviceServiceImplementation single instance>
```

6.1.2.3.2 CIM Requirements

See CIM_DeviceServiceImplementation in the “CIM Elements” section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.1.2.3.3 Behavior Requirements

6.1.2.3.3.1 Preconditions

`$instanceA` represents the instance of CIM_PlatformWatchdogService which is referenced by CIM_DeviceServiceImplementation.

`$instanceB` represents the instance of a concrete subclass of CIM_LogicalDevice which is referenced by CIM_DeviceServiceImplementation.

6.1.2.3.3.2 Pseudo Code

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

6.2 CIM_ElementCapabilities

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_ElementCapabilities

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

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

6.2.1 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.

6.2.2 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.

6.2.2.1 Show Command Form for a Single Instance – CIM_PlatformWatchdogService Reference

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

6.2.2.1.1 Command Form

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

6.2.2.1.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *Platform Watchdog 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 a CIM_PlatformWatchdogService, which is referenced by CIM_ElementCapabilities.

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

6.2.2.1.3.2 Pseudo Code

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

6.2.2.2 Show Command Form for Multiple Instances – CIM_PlatformWatchdogServiceCapabilities Reference

This command form is used to show multiple instances of CIM_ElementCapabilities. This command form corresponds to the `show` command issued against instances of CIM_ElementCapabilities where only one reference is specified and the reference is to the instance of CIM_PlatformWatchdogServiceCapabilities

6.2.2.2.1 Command Form

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

6.2.2.2.2 CIM Requirements

See CIM_ElementCapabilities in the “CIM Elements” section of the *Platform Watchdog 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 a CIM_PlatformWatchdogServiceCapabilties, which is referenced by CIM_ElementCapabilities.

6.2.2.2.3.2 Pseudo Code

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

6.2.2.3 Show a Single Instance Target – Both References

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

6.2.2.3.1 Command Form

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

6.2.2.3.2 CIM Requirements

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

6.2.2.3.3 Behavior Requirements

6.2.2.3.3.1 Preconditions

`$instanceA` represents the instance of a CIM_PlatformWatchdogService which is referenced by the CIM_ElementCapabilities.

`$instanceB` represents the instance of a CIM_PlatformWatchdogServiceCapabilities which is referenced by the CIM_ElementCapabilities.

6.2.2.3.3.2 Pseudo Code

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

6.3 CIM_HostedService

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 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_HostedService

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
rset	Not supported	
set	Not supported	
show	Shall	See 6.3.2.
start	Not supported	
stop	Not supported	

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

6.3.1 Ordering of Results

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

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

6.3.2 Show

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

6.3.2.1 Show Command Form for Multiple Instances – CIM_ComputerSystem Reference

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

6.3.2.1.1 Command Form

```
show <CIM_HostedService multiple objects>
```

6.3.2.1.2 CIM Requirements

See CIM_HostedService in the "CIM Elements" section of the *Platform Watchdog 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_ComputerSystem, which is referenced by CIM_HostedService.

6.3.2.1.3.2 Pseudo Code

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

6.3.2.2 Show Command Form for a Single Instance – CIM_PlatformWatchdogService Reference

This command form is to show a single instance of CIM_HostedService. The command form corresponds to the `show` verb issued against instances of CIM_HostedService where only one reference is specified and the reference is to an instance of CIM_PlatformWatchdogService.

6.3.2.2.1 Command Form

```
show <CIM_HostedService single object>
```

6.3.2.2.2 CIM Requirements

See CIM_HostedService in the "CIM Elements" section of the *Platform Watchdog 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_PlatformWatchdogService, which is referenced by CIM_HostedService.

6.3.2.2.3.2 Pseudo Code

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

6.3.2.3 Show Command Form for a Single Instance – Both References

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

6.3.2.3.1 Command Form

```
show <CIM_HostedService single object>
```

6.3.2.3.2 CIM Requirements

See CIM_HostedService in the “CIM Elements” section of the *Platform Watchdog 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_ComputerSystem through CIM_HostedService association.

`$instanceB` represents the other instance of CIM_PlatformWatchdogService which is referenced by CIM_HostedService.

6.3.2.3.3.2 Pseudo Code

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

6.4 CIM_ServiceAffectsElement

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_ServiceAffectsElement

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

Command Verb	Requirement	Comments
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`, `exit`, `load`, `reset`, `set`, `start`, and `stop`.

6.4.1 Ordering of Results

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

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

6.4.2 Show

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

6.4.2.1 Show Command Form for Multiple Instances – CIM_ComputerSystem Reference

This command form is used to show multiple instances of CIM_ComputerSystem. This command form corresponds to the `show` command issued against instances of CIM_ServiceAffectsElement 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_ServiceAffectsElement multiple instances>
```

6.4.2.1.2 CIM Requirements

See CIM_ServiceAffectsElement in the “CIM Elements” section of the *Platform Watchdog 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_ServiceAffectsElement.

6.4.2.1.3.2 Pseudo Code

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

6.4.2.2 Show Command Form for a Single Instance – CIM_LogicalElement Reference

This command form is used to show a single instance of CIM_ServiceAffectsElement. This command form corresponds to the `show` command issued against instances of CIM_ServiceAffectsElement where only one reference is specified and the reference is to an instance of a concrete subclass of CIM_LogicalElement.

6.4.2.2.1 Command Form

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

6.4.2.2.2 CIM Requirements

See CIM_ServiceAffectsElement in the "CIM Elements" section of the *Platform Watchdog 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 a concrete subclass of CIM_LogicalElement, which is referenced by CIM_ServiceAffectsElement.

6.4.2.2.3.2 Pseudo Code

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

6.4.2.3 Show Command Form for Multiple Instance – CIM_PlatformWatchdogService Reference

This command form is used when to show multiple instances of CIM_ServiceAffectsElement. This command form corresponds to the `show` command issued against instances of CIM_ServiceAffectsElement where only one reference is specified and the reference is to an instance of CIM_PlatformWatchdogService.

6.4.2.3.1 Command Form

```
show <CIM_ServiceAffectsElement multiple instances>
```

6.4.2.3.2 CIM Requirements

See CIM_ServiceAffectsElement in the "CIM Elements" section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.4.2.3.3 Behavior Requirements

6.4.2.3.3.1 Preconditions

`$instance` represents the instance of CIM_PlatformWatchdogService, which is referenced by CIM_ServiceAffectsElement.

6.4.2.3.3.2 Pseudo Code

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

6.4.2.4 Show Command Form for a Single Instance – Both References

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

6.4.2.4.1 Command Form

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

6.4.2.4.2 CIM Requirements

See CIM_ServiceAffectsElement in the “CIM Elements” section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.4.2.4.3 Behavior Requirements

6.4.2.4.3.1 Preconditions

`$instanceA` represents the instance of CIM_PlatformWatchdogService, which is referenced by CIM_ServiceAffectsElement.

`$instanceB` represents the instance of CIM_ComputerSystem or a concrete subclass of CIM_LogicalElement which is referenced by CIM_ServiceAffectsElement.

6.4.2.4.3.2 Pseudo Code

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

6.5 CIM_ServiceAvailableToElement

The `cd`, `exit`, `help`, and `version` verbs shallshall be supported as described in DSP0216.

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

Table 5 – Command Verb Requirements for CIM_ServiceAvailableToElement

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

6.5.1 Ordering of Results

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

- Results for CIM_ServiceAvailableToElement 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_ServiceAvailableToElement. Implementations shall support the use of the `show` verb with CIM_ServiceAvailableToElement.

6.5.2.1 Show Command Form for a Single Instance – CIM_LogicalElement Reference

This command form is used to show a single instance of CIM_ServiceAvailableToElement. This command form corresponds to the `show` command issued against instances of CIM_ServiceAvailableToElement where only one reference is specified and the reference is to an instance of a concrete subclass of CIM_LogicalElement.

6.5.2.1.1 Command Form

```
show <CIM_ServiceAvailableToElement single instance>
```

6.5.2.1.2 CIM Requirements

See CIM_ServiceAvailableToElement in the “CIM Elements” section of the *Platform Watchdog 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 a concrete subclass of CIM_LogicalElement, which is referenced by CIM_ServiceAvailableToElement.

6.5.2.1.3.2 Pseudo Code

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

6.5.2.2 Show Command Form for a Single Instance – CIM_PlatformWatchdogService Reference

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

6.5.2.2.1 Command Form

```
show <CIM_ServiceAvailableToElement single instance>
```

6.5.2.2.2 CIM Requirements

See CIM_ServiceAvailableToElement in the “CIM Elements” section of the *Platform Watchdog 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_PlatformWatchdogService, which is referenced by CIM_ServiceAvailableToElement.

6.5.2.2.3.2 Pseudo Code

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

6.5.2.3 Show Command Form for a Single Instance – Both References

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

6.5.2.3.1 Command Form

```
show <CIM_ServiceAvailableToElement single instance>
```

6.5.2.3.2 CIM Requirements

See CIM_ServiceAvailableToElement in the “CIM Elements” section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.5.2.3.3 Behavior Requirements

6.5.2.3.3.1 Preconditions

`$instanceA` represents the instance of CIM_PlatformWatchdogService, which is referenced by CIM_ServiceAvailableToElement.

`$instanceB` represents the instance of a concrete subclass of CIM_LogicalElement which is referenced by CIM_ServiceAvailableToElement.

6.5.2.3.3.2 Pseudo Code

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

6.6 CIM_PlatformWatchdogService

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 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_PlatformWatchdogService

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

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

6.6.1 Ordering of Results

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

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

6.6.2 Set

This section describes how to implement the `set` verb when it is applied to an instance of CIM_PlatformWatchdogService. The `set` verb is used to set properties on an instance of CIM_PlatformWatchdogService.

Implementations may support the use of the `set` verb with CIM_PlatformWatchdogService.

6.6.2.1 General Usage of Set for a Single Property

This command form corresponds to the general usage of the set verb to modify a single property of a target instance. The setting of a single property shall be deterministic.

The requirements 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 *Platform Watchdog Profile*.

6.6.2.1.1 Command Form

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

6.6.2.1.2 CIM Requirements

See CIM_PlatformWatchdogService in the “CIM Elements” section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.6.2.1.3 Behavior Requirements

```
$instance=<CIM_PlatformWatchdogService single object>
#propertyNames[]  = <propertname>
```

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

6.6.2.2 General Usage of Set for Multiple Properties

This command form corresponds to the general usage of the `set` verb to modify multiple properties of a target instance. The setting of multiple properties may be deterministic.

The requirements 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 *Platform Watchdog Profile*.

6.6.2.2.1 Command Form

```
set <CIM_PlatformWatchdogService single object> <propertyname1>=<propertyvalue1>
   <propertynameN>=<propertyvalueN>
```

6.6.2.2.2 CIM Requirements

See CIM_PlatformWatchdogService in the “CIM Elements” section of the *Platform Watchdog 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_PlatformWatchdogService.

6.6.2.2.3.2 Pseudo Code

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

6.6.3 Show

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

6.6.3.1 Show Command Form for a Single Instance

This command form is to show a single instance of CIM_PlatformWatchdogService.

6.6.3.1.1 Command Form

```
show <CIM_PlatformWatchdogService single object>
```

6.6.3.1.2 CIM Requirements

See CIM_PlatformWatchdogService in the “CIM Elements” section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.6.3.1.3 Behavior Requirements

6.6.3.1.3.1 Preconditions

`$instance` represents the instance of CIM_PlatformWatchdogService.

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

`#propertylist[]` is an array of mandatory non-key property names.

6.6.3.1.3.2 Pseudo Code

```
if (false != #all) { #propertylist[] = NULL; }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.6.3.2 Show Command Form for Multiple Instances

This command form is used to show multiple instances of CIM_PlatformWatchdogService.

6.6.3.2.1 Command Form

```
show <CIM_PlatformWatchdogService multiple objects>
```

6.6.3.2.2 CIM Requirements

See CIM_PlatformWatchdogService in the “CIM Elements” section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.6.3.2.3 Behavior Requirements

6.6.3.2.3.1 Preconditions

`$containerInstance` represents the instance of CIM_ComputerSystem which represents the container system and is associated with the target instance of CIM_PlatformWatchdogService through the CIM_HostedService association.

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

`#propertylist[]` is an array of mandatory non-key property names.

6.6.3.2.3.2 Pseudo Code

```
if (false != #all) { #propertylist[] = NULL; }
&smShowInstances ( “CIM_PlatformWatchdogService”, “CIM_HostedService”,
   $containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.6.4 Start

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

6.6.4.1 Start Command Form for a Single Instance

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

6.6.4.1.1 Command Form

```
start <CIM_PlatformWatchdogService single object>
```

6.6.4.1.2 CIM Requirements

See CIM_PlatformWatchdogService in the “CIM Elements” section of the Platform Watchdog Profile for the list of mandatory properties.

6.6.4.1.3 Behavior Requirements

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

6.6.5 Stop

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

6.6.5.1 Stop Command Form for a Single Instance

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

6.6.5.1.1 Command Form

```
stop <CIM_PlatformWatchdogService single object>
```

6.6.5.1.2 CIM Requirements

See CIM_PlatformWatchdogService in the “CIM Elements” section of the Platform Watchdog Profile for the list of mandatory properties.

6.6.5.1.3 Behavior Requirements

```
$instance=<CIM_PlatformWatchdogService single object>
smRequestStateChange ( $instance.getObjectPath(), “Disabled” );
&smEnd;
```

6.7 CIM_PlatformWatchdogServiceCapabilities

The `cd`, `exit`, `help`, and `version` verbs shallshall be supported as described in DSP0216.

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

Table 7 – Command Verb Requirements for CIM_PlatformWatchdogServiceCapabilities

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

Command Verb	Requirement	Comments
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`, `exit`, `load`, `reset`, `set`, `start`, and `stop`.

6.7.1 Ordering of Results

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

- Results for CIM_PlatformWatchdogServiceCapabilities 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_PlatformWatchdogServiceCapabilities. The `show` verb is used to display information about instances of CIM_PlatformWatchdogServiceCapabilities.

Implementations shall support the use of the `show` verb with CIM_PlatformWatchdogServiceCapabilities.

6.7.2.1 Show Command Form for a Single Instance

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

6.7.2.1.1 Command Form

```
show <CIM_PlatformWatchdogServiceCapabilities single object>
```

6.7.2.1.2 CIM Requirements

See CIM_PlatformWatchdogServiceCapabilities in the “CIM Elements” section of the *Platform Watchdog 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_PlatformWatchdogServiceCapabilities.

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

`#propertylist[]` is an array of mandatory non-key property names.

6.7.2.1.3.2 Pseudo Code

```
if (false != #all) { #propertylist[] = NULL; }
&smShowInstance ( $instance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.7.2.2 Show Multiple Instances

This command form is used to show multiple instances of CIM_PlatformWatchdogServiceCapabilities.

6.7.2.2.1 Command Form

```
show <CIM_PlatformWatchdogServiceCapabilities multiple objects>
```

6.7.2.2.2 CIM Requirements

See CIM_PlatformWatchdogServiceCapabilities in the "CIM Elements" section of the *Platform Watchdog Profile* for the list of mandatory properties.

6.7.2.2.3 Behavior Requirements

6.7.2.2.3.1 Preconditions

`$containerInstance` represents the instance of CIM_ConcreteCollection with ElementName property that contains "Capabilities" and is associated with the targeted instances of CIM_PlatformWatchdogServiceCapabilities through the CIM_MemberOfCollection association.

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

`#propertylist[]` is an array of mandatory non-key property names.

6.7.2.2.3.2 Pseudo Code

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

ANNEX A
(informative)

Change Log

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