Document Number: DSP0801

Date: 2009-07-29

Version: 1.0.0

Command Line Protocol Service Profile SM CLP Command Mapping Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Tables

Foreword

The *Command Line Protocol Service Profile SM CLP Command Mapping Specification* (DSP0801) was prepared by the Server Management Working Group and the Physical Profiles Working Group of the DMTF.

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:

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

Introduction

This document defines the SM CLP mapping for CIM elements described in the *CLP Service 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 *CLP Service Profile* using CIM operations.

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

Command Line Protocol Service 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 *CLP Service Profile*.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved References

DMTF DSP0216, *SM CLP-to-CIM Common Mapping Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0216_1.0.pdf

DMTF DSP1005, *CLP Service Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1005_1.0.pdf

SNIA, *Storage Management Initiative Specification (SMI-S) 1.1.0*, November 2005,
http://www.snia.org/tech_activities/standards/curr_standards/smi/

2.2 Other References

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

3 Terms and Definitions

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

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

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

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

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

3.5
may
indicates a course of action permissible within the limits of the document

3.6
need not
indicates a course of action permissible within the limits of the document

3.7
optional
indicates a course of action permissible within the limits of the document

3.8
shall
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.9
shall not
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.10
should
indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.11
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1
CIM
Common Information Model

4.2
CLP
Command Line Protocol

4.3
DMTF
Distributed Management Task Force

4.4
IETF
Internet Engineering Task Force

4.5
SM
Server Management

4.6
SMI-S
Storage Management Initiative Specification

4.7
SNIA
Storage Networking Industry Association

4.8
UFsT
User Friendly selection Tag

5 Recipes

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

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

6 Mappings

The following sections detail the mapping of CLP verbs to CIM Operations for each CIM class defined in the *CLP Service 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_BindsTo

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_BindsTo

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

- Results for CIM_BindsTo 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_BindsTo. Implementations shall support the use of the `show` verb with CIM_BindsTo.

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

6.1.2.1 Show a Single Instances – CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint

This command form is for the `show` verb applied to multiple instances. This command form corresponds to the `show` command issued against CIM_BindsTo where only one reference is specified and the reference is to an instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint.

6.1.2.1.1 Command Form

```
show <CIM_BindsTo multiple instances>
```

6.1.2.1.2 CIM Requirements

See CIM_BondsTo in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.1.2.1.3 Behavior Requirements

6.1.2.1.3.1 Preconditions

```
$instance contains the instance of CIM_SSHProtcolEndpoint or
   CIM_TelnetProtocolEndpoingwhich is referenced by CIM_BindsTo
```

6.1.2.1.3.2 Pseudo Code

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

6.1.2.2 Show a Single Instance – CIM_CLPProtocolEndpoint Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_BindsTo where the reference specified is to an instance of CIM_CLPProtocolEndpoint. A single instance of CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint is associated with each instance of CIM_CLPProtocolEndpoint. Therefore, a single instance will be returned.

6.1.2.2.1 Command Form

`show <CIM_BindsTo `*`single instance`*`>`

6.1.2.2.2 CIM Requirements

See the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.1.2.2.3 Behavior Requirements

6.1.2.2.3.1 Preconditions

```
$instance contains the instance of CIM_CLPProtocolEndpoint which is referenced by
   CIM_BindsTo
```

6.1.2.2.3.2 Pseudo Code

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

6.1.2.3 Show a Single Instance – Both References

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_BindsTo where a reference to CIM_CLPProtocolEndpoint and a reference to CIM_SSHProtocolEndpoint or CIM_TelnetProtocolEndpoint are specified and therefore the desired instance is unambiguously identified.

6.1.2.3.1 Command Form

`show <CIM_BindsTo `*`single instance`*`>`

6.1.2.3.2 CIM Requirements

See the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.1.2.3.3 Behavior Requirements

6.1.2.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_CLPProtocolEndpoint which is referenced by
```

```
   CIM_BindsTo
2) $instanceB contains the instance of CIM_ProtocolEndpoint which is referenced by
   CIM_BindsTo.
```

6.1.2.3.3.2 Pseudo Code

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

6.2 CIM_CLPCapabilities

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

Table 2 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 2 is for informational purposes only; in case of a conflict between Table 2 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 2.

Table 2 – Command Verb Requirements for CIM_CLPCapabilities

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

6.2.1 Ordering of Results

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

- Results for CIM_CLPCapabilities 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_CLPCapabilities. Implementations shall support the use of the `show` verb with CIM_CLPCapabilities.

The `show` verb is used to display information about an instance or instances of the CIM_CLPCapabilities class.

6.2.2.1 Show a Single Instance

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

6.2.2.1.1 Command Form

```
show <CIM_CLPCapabilities single instance>
```

6.2.2.1.2 CIM Requirements

See CIM_CLPCapabilities in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.2.2.1.3 Behavior Requirements

6.2.2.1.3.1 Preconditions

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

6.2.2.1.3.2 Pseudo Code

```
$instance=<CIM_CLPCapabilities Single Instance>
#propertylist[] = null;
if ( false == #all) {
   #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
&smEnd;
```

6.3 CIM_CLPProtocolEndpoint

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_CLPProtocolEndpoint

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.3.2.
show	Shall	See 6.3.3.
start	Not supported	
stop	May	See 6.3.4.

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

6.3.1 Ordering of Results

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

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

6.3.2 Set

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

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

6.3.2.1 General Usage of Set for a Single Property

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.3.2.1.1 Command Form

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

6.3.2.1.2 CIM Requirements

See CIM_CLPProtocolEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of modifiable properties.

6.3.2.1.3 Behavior Requirements

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

6.3.2.2 General Usage of Set for Multiple Properties

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.3.2.2.1 Command Form

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

6.3.2.2.2 CIM Requirements

See CIM_CLPProtocolEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.3.2.2.3 Behavior Requirements

```
$instance = <CIM_CLPProtocolEndpoint Single Instance>
#propertyNames[] = {<propertyname>};

for #i < n
{
   #propertyNames[#i] = <propertname#i>
   #propertyValues[#i] = <propertyvalue#i>
}

&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.3.3 Show

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

The `show` verb is used to display information about a CLP session.

6.3.3.1 Show a Single Instance

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

6.3.3.1.1 Command Form

```
show <CIM_CLPProtocolEndpoint single instance>
```

6.3.3.1.2 CIM Requirements

See CIM_CLPProtocolEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.3.3.1.3 Behavior Requirements

6.3.3.1.3.1 Preconditions

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

6.3.3.1.3.2 Pseudo Code

```
$instance=<CIM_CLPProtocolEndpoint Single Instance>
#propertylist[] = null;
if ( false == #all) {
   #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
```

6.3.3.2 Show Multiple Instances Scoped by System

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

6.3.3.2.1 Command Form

```
show <CIM_CLPProtocolEndpoint multiple instances>
```

6.3.3.2.2 CIM Requirements

See CIM_CLPProtocolEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.3.3.2.3 Behavior Requirements

6.3.3.2.3.1 Preconditions

```
1) $containerInstance contains the instance of CIM_ComputerSystem for which we are
   displaying scoped endpoints (CIM_CLPProtocolEndpoint instances). The CLP Service
   Profile requires that the CIM_CLPProtocolEndpoint instance be associated with its
   scoping system via an instance of the CIM_HostedAccessPoint association.
2) #all is true if the all option was specified with the command; otherwise, #all is
   false
```

6.3.3.2.3.2 Pseudo Code

```
#propertylist[] = null;
if (false == #all)  {
   #propertylist[] = {//all mandatory non-key properties }
}

&smShowInstances ( “CIM_CLPProtocolEndpoint”, “CIM_HostedAccessPoint",
$containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.3.4 Stop

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

The `stop` verb is used to terminate a CLP session.

6.3.4.1 Stop a Single Instance

This command form is for the `stop` verb applied to a single instance of CIM_CLPProtocolEndpoint. The lifecycle of a CLP session corresponds to the lifecycle of the CIM_CLPProtocolEndpoint which represents it. Therefore, stopping a CLP service corresponds to a deletion of the underlying instance.

6.3.4.1.1 Command Form

```
stop <CIM_CLPProtocolEndpoint single instance>
```

6.3.4.1.2 CIM Requirements

See CIM_CLPProtocolEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.3.4.1.3 Behavior Requirements

```
$instance=<CIM_CLPProtocolEndpoint Single Instance>
&smDeleteInstance($instance.getObjectPath());
&smEnd;
```

6.4 CIM_CLPSettingData

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

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

Table 4 – Command Verb Requirements for CIM_CLPSettingData

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.4.2.
show	Shall	See 6.4.3.
start	Not supported	
stop	Not supported	

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

6.4.1 Ordering of Results

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

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

6.4.2 Set

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

The `set` verb is used to modify configuration represented by an instance of CIM_CLPSettingData.

6.4.2.1 General Usage of Set for a Single Property

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.4.2.1.1 Command Form

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

6.4.2.1.2 CIM Requirements

See CIM_CLPSettingData in the “CIM Elements” section of the *CLP Service Profile* for the list of modifiable properties.

6.4.2.1.3 Behavior Requirements

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

6.4.2.2 General Usage of Set for Multiple Properties

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.4.2.2.1 Command Form

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

6.4.2.2.2 CIM Requirements

See CIM_CLPSettingData in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.4.2.2.3 Behavior Requirements

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

6.4.3 Show

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

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

6.4.3.1 Show a Single Instance

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

6.4.3.1.1 Command Form

```
show <CIM_CLPSettingData single instance>
```

6.4.3.1.2 CIM Requirements

See CIM_CLPSettingData in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.4.3.1.3 Behavior Requirements

6.4.3.1.3.1 Preconditions

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

6.4.3.1.3.2 Pseudo Code

```
$instance=<CIM_CLPSettingData Single Instance>
&lShowPEndpoint($instance, #all);
&smEnd;
```

6.4.3.2 Show Multiple Instances Scoped by ConcreteCollection

This command form is for the `show` verb applied to multiple instances of CIM_CLPSettingData. This command form corresponds to UFsT-based selection within an instance of CIM_ConcreteCollection.

6.4.3.2.1 Command Form

```
show <CIM_CLPSettingData multiple instances>
```

6.4.3.2.2 CIM Requirements

See CIM_CLPSettingData in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.4.3.2.3 Behavior Requirements

6.4.3.2.3.1 Preconditions

```
1) $containerInstance contains the instance of CIM_ConcreteCollection for which we are
   displaying contained CIM_CLPSettingData instances. The SMASH Collections Profile
   requires that the CIM_CLPSettingData instances be aggregated into an addressing
   collection via CIM_MemberOfCollection.
2) #all is true if the all option was specified with the command; otherwise, #all is
   false
```

6.4.3.2.3.2 Pseudo Code

```
#propertylist[] = null;
//this property list will match the property list in lShowPEndpoint()
if (false == #all)  {
   #propertylist[] = {//all mandatory non-key properties }
}

&smShowInstances ( “CIM_CLPSettingData”, “CIM_MemberOfCollection",
$containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.5 CIM_ConcreteJob

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

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

Table 5 – Command Verb Requirements for CIM_ConcreteJob

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	May	See 6.5.2.
show	Shall	See 6.5.3.
start	Not supported	
stop	May	See 6.5.4.

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

6.5.1 Ordering of Results

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

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

6.5.2 Set

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

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

6.5.2.1 General Usage of Set for a Single Property

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.5.2.1.1 Command Form

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

6.5.2.1.2 CIM Requirements

See CIM_ConcreteJob in the “CIM Elements” section of the *CLP Service Profile* for the list of modifiable properties.

6.5.2.1.3 Behavior Requirements

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

6.5.2.2 General Usage of Set for Multiple Properties

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

The requirement for supporting modification of a property using this command form shall be equivalent to the requirement for supporting modification of the property using the ModifyInstance operation as defined in the *CLP Service Profile*.

6.5.2.2.1 Command Form

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

6.5.2.2.2 CIM Requirements

See CIM_ConcreteJob in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.5.2.2.3 Behavior Requirements

```
$instance = <CIM_ConcreteJob Single Instance>
#propertyNames[] = {<propertyname>};

for #i < n
{
   #propertyNames[#i] = <propertname#i>
   #propertyValues[#i] = <propertyvalue#i>
}

```

```
&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.5.3 Show

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

The `show` verb is used to display information about the CLP Service's Job Queue.

6.5.3.1 Show a Single Instance

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

6.5.3.1.1 Command Form

```
show <CIM_ConcreteJob single instance>
```

6.5.3.1.2 CIM Requirements

See CIM_ConcreteJob in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.5.3.1.3 Behavior Requirements

6.5.3.1.3.1 Preconditions

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

6.5.3.1.3.2 Pseudo Code

```
$instance=<CIM_ConcreteJob Single Instance>
#propertylist[] = null;
if ( false == #all) {
   #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
```

6.5.3.2 Show Multiple Instances Scoped by JobQueue

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

6.5.3.2.1 Command Form

```
show <CIM_ConcreteJob multiple instances>
```

6.5.3.2.2 CIM Requirements

See CIM_ConcreteJob in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.5.3.2.3 Behavior Requirements

6.5.3.2.3.1 Preconditions

```
1) $containerInstance contains the instance of CIM_JobQueue for which we are
   displaying scoped jobs (CIM_ConcreteJob instances). The CLP Service Profile
```

```
   requires that the CIM_ConcreteJob instance be associated with its scoping job queue
   via an instance of the CIM_JobDestinationJobs association.
2) #all is true if the all option was specified with the command; otherwise, #all is
   false
```

6.5.3.2.3.2 Pseudo Code

```
#propertylist[] = null;
if (false == #all)  {
   #propertylist[] = {//all mandatory non-key properties }
}

&smShowInstances ( “CIM_ConcreteJob”, “CIM_JobDestinationJobs",
   $containerInstance.getObjectPath(), #propertylist[] );
   &smEnd;
```

6.5.4 Stop

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

The `stop` verb is used to terminate or kill a CLP operation.

6.5.4.1 Stop a Single Instance

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

6.5.4.1.1 Command Form

```
stop [-f] <CIM_ProtocolService single instance>
```

6.5.4.1.2 CIM Requirements

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

6.5.4.1.3 Behavior Requirements

6.5.4.1.3.1 Preconditions

```
#force is true if force option was specified; otherwise, #force is false
if (#force)  {
   #requestedState = 5;//kill
}
else {
   #requestedstate = 4;//terminate
}
   $instance=<CIM_ProtocolService Single Instance>
&smRequestStateChange ( $instance.getObjectPath(), #requestedState );
smEnd;
```

6.6 CIM_ElementCapabilities

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

Table 6 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 6 is for informational purposes only; in case of a conflict between Table 6 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 6.

Table 6 – Command Verb Requirements for CIM_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 6.6.2.
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_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.

6.6.2 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.

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

6.6.2.1 Show Multiple Instances – CIM_CLPCapabilities Reference

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 the reference specified is to an instance of CIM_CLPCapabilities. Multiple instances of CIM_ProtocolService can be associated with each instance of a CIM_CLPCapabilities.

6.6.2.1.1 Command Form

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

6.6.2.1.2 CIM Requirements

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

6.6.2.1.3 Behavior Requirements

6.6.2.1.3.1 Preconditions

```
$instance contains the instance of CIM_CLPCapabilities which is referenced by
CIM_ElementCapabilities
```

6.6.2.1.3.2 Pseudo Code

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

6.6.2.2 Show a Single Instance – CIM_ProtocolService Reference

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 the reference specified is to an instance of CIM_ProtocolService. A single instance of CIM_CLPCapabilities is associated with each instance of CIM_ProtocolService. Therefore, a single instance will be returned.

6.6.2.2.1 Command Form

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

6.6.2.2.2 CIM Requirements

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

6.6.2.2.3 Behavior Requirements

6.6.2.2.3.1 Preconditions

`$instance` contains the instance of CIM_ProtocolService which is referenced by CIM_ElementCapabilities

6.6.2.2.3.2 Pseudo Code

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

6.6.2.3 Show a Single Instance – Both References

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

6.6.2.3.1 Command Form

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

6.6.2.3.2 CIM Requirements

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

744 #### 6.6.2.3.3 Behavior Requirements

745 ##### 6.6.2.3.3.1 Preconditions

```
746 1) $instanceA contains the instance of CIM_CLPCapabilities  which is referenced by
747    CIM_ElementCapabilities
748 2) $instanceB contains the instance of CIM_ProtocolService which is  referenced by
749    CIM_ElementCapabilities.
```

750 ##### 6.6.2.3.3.2 Pseudo Code

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

754 ## 6.7 CIM_ElementSettingData

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

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

761 **Table 7 – Command Verb Requirements for CIM_ElementSettingData**

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

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

764 ### 6.7.1 Ordering of Results

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

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

768 ### 6.7.2 Set

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

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

773 6.7.2.1 Set of IsNext

774 The IsNext property is the only property of CIM_ElementSettingData which can be modified directly via
775 the `set` verb.

776 6.7.2.1.1 Command Form

777 **`set <CIM_ElementSettingData single instance> IsNext=<propertyvalue>`**

778 6.7.2.1.2 CIM Requirements

779 See CIM_ElementSettingData in the “CIM Elements” section of the *CLP Service Profile* for the list of
780 mandatory properties.

781 6.7.2.1.3 Behavior Requirements

```
782 $instance = <CIM_ElementSettingData Single Instance>
783 #propertyNames[] = {"IsNext"};
784 #propertyValues[] = {<propertyvalue>};
785 &smSetInstance($instance, #propertyNames[], #propertyValues[]);
786 &smEnd;
```

787 6.7.3 Show

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

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

793 6.7.3.1 Show Multiple Instances – CIM_CLPSettingData and CIM_CLPProtocolEndpoint

794 This command form corresponds to the `show` command issued against CIM_ElementSettingData where
795 the reference specified is to an instance of CIM_CLPSettingData. Note that when an instance of
796 CIM_CLPSettingData is associated with an instance of CIM_CLPProtocolEndpoint, the IsCurrent property
797 is the mandatory property.

798 6.7.3.1.1 Command Form

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

800 6.7.3.1.2 CIM Requirements

801 See CIM_ElementSettingData in the “CIM Elements” section of the *CLP Service Profile* for the list of
802 mandatory properties.

803 6.7.3.1.3 Behavior Requirements

804 6.7.3.1.3.1 Preconditions

```
805 1) $instance contains the instance of CIM_CLPSettingData which is referenced by
806    CIM_ElementSettingData
807 2) #all is true if the all option was specified
```

6.7.3.1.3.2 Pseudo Code

```
#propertylist = NULL;
if (false == #all)  {
   #propertylist = { "IsCurrent" };
}
   &smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getObjectPath(),
   #propertylist[] );
   smEnd;
```

6.7.3.2 Show Multiple Instances – CIM_CLPProtocolEndpoint Reference

This command form corresponds to the `show` command issued against CIM_ElementSettingData where the reference specified is to an instance of CIM_CLPProtocolEndpoint. Note that when an instance of CIM_CLPSettingData is associated with an instance of CIM_CLPProtocolEndpoint, the IsCurrent property is the mandatory property.

6.7.3.2.1 Command Form

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

6.7.3.2.2 CIM Requirements

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

6.7.3.2.3 Behavior Requirements

6.7.3.2.3.1 Preconditions

```
1) $instance contains the instance of CIM_CLPProtocolEndpoint which is referenced by
   CIM_ElementSettingData
2) #all is true if the all option was specified
```

6.7.3.2.3.2 Pseude Code

```
#propertylist = NULL;
if (false == #all)  {
   #propertylist = { "IsCurrent" };
}
&smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getObjectPath(),
#propertylist[] );
smEnd;
```

6.7.3.3 Show a Single Instance – CIM_CLPSettingData and CIM_CLPProtocolEndpoint

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

6.7.3.3.1 Command Form

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

6.7.3.3.2 CIM Requirements

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

6.7.3.3.3 Behavior Requirements

6.7.3.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_CLPSettingData  which is referenced by
   CIM_ElementSettingData
2) $instanceB contains the instance of CIM_CLPProtocolEndpoint which is referenced by
   CIM_ElementSettingData.
3) #all is true if the all option was specified
```

6.7.3.3.3.2 Pseudo Code

```
#propertylist = NULL;
if (false == #all)  {
   #propertylist = { "IsCurrent" };
}
&smShowAssociationInstance ( “CIM_ElementSettingData”,
   $instanceA.getObjectPath(),$instanceB.getObjectPath(),
#propertylist[] );
smEnd;
```

6.7.3.4 Show Multiple Instances – CIM_CLPSettingData and CIM_ProtocolService

This command form corresponds to the `show` command issued against CIM_ElementSettingData where the reference specified is to an instance of CIM_CLPSettingData. Note that when an instance of CIM_CLPSettingData is associated with an instance of CIM_ProtocolService, the IsNext and IsDefault properties are mandatory.

6.7.3.4.1 Command Form

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

6.7.3.4.2 CIM Requirements

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

6.7.3.4.3 Behavior Requirements

6.7.3.4.3.1 Preconditions

```
1) $instance contains the instance of CIM_CLPSettingData which is referenced by
   CIM_ElementSettingData
2) #all is true if the all option was specified
```

6.7.3.4.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)  {
   #propertylist = { "IsNext", "IsDefault" };
}
   &smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getObjectPath(),
```

```
   #propertylist[] );
   smEnd;
```

6.7.3.5 Show Multiple Instances – CIM_ProtocolService Reference

This command form corresponds to the `show` command issued against CIM_ElementSettingData where the reference specified is to an instance of CIM_ProtocolService. Note that when an instance of CIM_CLPSettingData is associated with an instance of CIM_ProtocolService, the IsNext and IsDefault properties are mandatory.

6.7.3.5.1 Command Form

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

6.7.3.5.2 CIM Requirements

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

6.7.3.5.3 Behavior Requirements

6.7.3.5.3.1 Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_ElementSettingData
2) #all is true if the all option was specified
```

6.7.3.5.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)  {
   #propertylist = { "IsNext", "IsDefault" };
}
&smShowAssociationInstances ( “CIM_ElementSettingData”, $instance.getObjectPath(),
#propertylist[] );
smEnd;
```

6.7.3.6 Show a Single Instance – CIM_CLPSettingData and CIM_ProtocolService

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

6.7.3.6.1 Command Form

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

6.7.3.6.2 CIM Requirements

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

6.7.3.6.3 Behavior Requirements

6.7.3.6.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_CLPSettingData  which is referenced by
   CIM_ElementSettingData
```

```
2) $instanceB contains the instance of CIM_ProtocolService which is  referenced by
   CIM_ElementSettingData.
3) #all is true if the all option was specified
```

6.7.3.6.3.2 Pseudo Code

```
#propertylist[] = NULL;
if (false == #all)  {
   #propertylist = { "IsNext", "IsDefault" };
}
&smShowAssociationInstance ( “CIM_ElementSettingData”,
   $instanceA.getObjectPath(),$instanceB.getObjectPath(), #propertylist[] );
smEnd;
```

6.8 CIM_Error

CIM_Error is utilized as a template for embedded instances. Instances of the class are not exposed by the underlying profile. Therefore, the usage of any CLP verbs shall not be supported for instances of CIM_Error.

6.9 CIM_HostedAccessPoint

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

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

Table 8 – Command Verb Requirements for CIM_HostedAccessPoint

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.9.2.
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_HostedAccessPoint, implementations shall utilize the following algorithm to produce the natural (that is, default) ordering:

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

6.9.2 Show

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

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

6.9.2.1 Show Multiple Instances

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

6.9.2.2 Command Form

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

6.9.2.2.1 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.9.2.2.2 Behavior Requirements

6.9.2.2.2.1 Preconditions

```
1) $instance contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedAccessPoint
```

6.9.2.2.2.2 Pseudo Code

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

6.9.2.3 Show a Single Instance – CIM_CLPProtocolEndpoint Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_HostedAccessPoint where the reference specified is to an instance of CIM_CLPProtocolEndpoint. A single instance will be returned.

6.9.2.3.1 Command Form

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

6.9.2.3.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.9.2.3.3 Behavior Requirements

6.9.2.3.3.1 Preconditions

```
1) $instance contains the instance of CIM_ProtocolEndpoint that is referenced by
   CIM_HostedAccessPoint
```

6.9.2.3.3.2 Pseudo Code

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

6.9.2.4 Show a Single Instance – 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_HostedAccessPoint where both references are specified and therefore the desired instance is unambiguously identified.

6.9.2.4.1 Command Form

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

6.9.2.4.2 CIM Requirements

See CIM_HostedAccessPoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.9.2.4.3 Behavior Requirements

6.9.2.4.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedAccessPoint
2) $instanceB contains the instance of CIM_CLPProtocolEndpoint that is referenced by
   CIM_HostedAccessPoint
```

6.9.2.4.3.2 Pseudo Code

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

6.10 CIM_HostedJobDestination

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

Table 9 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 9 is for informational purposes only; in case of a conflict between Table 9 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 9.

Table 9 – Command Verb Requirements for CIM_HostedJobDestination

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.10.2.

Command Verb	Requirement	Comments
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`.

6.10.1 Ordering of Results

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

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

6.10.2 Show

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

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

6.10.2.1 Show a Single Instance – CIM_JobQueue Reference

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

6.10.2.1.1 Command Form

```
show <CIM_HostedJobDestination single instance>
```

6.10.2.1.2 CIM Requirements

See CIM_HostedJobDestination in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.10.2.1.3 Behavior Requirements

6.10.2.1.3.1 Preconditions

```
1) $instance contains the instance of CIM_JobQueue
   which is referenced by CIM_HostedJobDestination
   &smShowAssociationInstances ( “CIM_HostedJobDestination”,
   $instance.getObjectPath() );
   smEnd;
```

6.10.2.2 Show a Single Instance – CIM_ComputerSystem Reference

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

6.10.2.2.1 Command Form

```
show <CIM_HostedJobDestination single instance>
```

6.10.2.2.2 CIM Requirements

See CIM_HostedJobDestination in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.10.2.2.3 Behavior Requirements

6.10.2.2.3.1 Preconditions

```
1) $instance contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedJobDestination
```

6.10.2.2.3.2 Pseudo Code

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

6.10.2.3 Show a Single Instance – Both References

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

6.10.2.3.1 Command Form

```
show <CIM_HostedJobDestination single instance>
```

6.10.2.3.2 CIM Requirements

See CIM_HostedJobDestination in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.10.2.3.3 Behavior Requirements

6.10.2.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedJobDestination
2) $instanceB contains the instance of CIM_JobQueue which is referenced by
   CIM_HostedJobDestination.
```

6.10.2.3.3.2 Pseudo Code

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

6.11 CIM_HostedService

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

Table 10 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 10 is for informational purposes only; in case of a conflict between Table 10 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 10.

1090 **Table 10 – Command Verb Requirements for CIM_HostedService**

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.11.2.
start	Not supported	
stop	Not supported	

1091 No mapping is defined for the following verbs for the specified target: `create`, delete, `dump`, `load`,
1092 `reset`, `set`, `start`, and `stop`.

1093 ### 6.11.1 Ordering of Results

1094 When results are returned for multiple instances of CIM_HostedService, implementations shall utilize the
1095 following algorithm to produce the natural (that is, default) ordering:

1096 - Results for CIM_HostedService are unordered; therefore, no algorithm is defined.

1097 ### 6.11.2 Show

1098 This section describes how to implement the `show` verb when applied to an instance of
1099 CIM_HostedService. Implementations shall support the use of the `show` verb with CIM_HostedService.

1100 The `show` command is used to display information about the CIM_HostedService instance or instances.

1101 #### 6.11.2.1 Show Multiple Instances

1102 This command form is for the `show` verb applied to multiple instances. This command form corresponds
1103 to a `show` command issued against CIM_HostedService where only one reference is specified and the
1104 reference is to an instance of CIM_ComputerSystem.

1105 ##### 6.11.2.1.1 Command Form

1106 **`show <CIM_HostedService multiple instances>`**

1107 ##### 6.11.2.1.2 CIM Requirements

1108 See CIM_HostedService in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory
1109 properties.

1110 ##### 6.11.2.1.3 Behavior Requirements

1111 ###### 6.11.2.1.3.1 Preconditions

1112 `1) $instance contains the instance of CIM_ComputerSystem which is referenced by`
1113 `CIM_HostedService`

6.11.2.1.3.2 Pseudo Code

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

6.11.2.2 Show a Single Instance – CIM_ProtocolService Reference

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

6.11.2.2.1 Command Form

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

6.11.2.2.2 CIM Requirements

See CIM_HostedService in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.11.2.2.3 Behavior Requirements

6.11.2.2.3.1 Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_HostedService
```

6.11.2.2.3.2 Pseudo Code

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

6.11.2.3 Show a Single Instance – Both References

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

6.11.2.3.1 Command Form

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

6.11.2.3.2 CIM Requirements

See CIM_HostedService in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.11.2.3.3 Behavior Requirements

6.11.2.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_ComputerSystem which is referenced by
   CIM_HostedService
2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
   CIM_HostedService.
```

6.11.2.3.3.2 Pseudo Code

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

6.12 CIM_JobDestinationJobs

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

Table 11 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 11 is for informational purposes only; in case of a conflict between Table 11 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 11.

Table 11 – Command Verb Requirements for CIM_JobDestinationJobs

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.12.2.
start	Not supported	
stop	Not supported	

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

6.12.1 Ordering of Results

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

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

6.12.2 Show

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

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

6.12.2.1 Show a Single Instance – CIM_ConcreteJob Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_JobDestinationJobs where the reference specified is to an instance of CIM_ConcreteJob. A single instance of CIM_JobQueue is associated with each instance of a CIM_ConcreteJob Therefore, a single instance will be returned.

6.12.2.1.1 Command Form

```
show <CIM_JobDestinationJobs single instance>
```

6.12.2.1.2 CIM Requirements

See CIM_JobDestinationJobs in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.12.2.1.3 Behavior Requirements

6.12.2.1.3.1 Preconditions

```
$instance contains the instance of CIM_ConcreteJob which is referenced by
   CIM_JobDestinationJobs
```

6.12.2.1.3.2 Pseudo Code

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

6.12.2.2 Show Multiple Instances – CIM_JobQueue Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_JobDestinationJobs where the reference specified is to an instance of CIM_JobQueue. A single instance of CIM_JobQueue is associated with multiple instances of CIM_ConcreteJob. Therefore, multiple instances may be returned.

6.12.2.2.1 Command Form

```
show <CIM_JobDestinationJobs multiple instances>
```

6.12.2.2.2 CIM Requirements

See CIM_JobDestinationJobs in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.12.2.2.3 Behavior Requirements

6.12.2.2.3.1 Preconditions

```
$instance contains the instance of CIM_JobQueue which is referenced by
   CIM_JobDestinationJobs
```

6.12.2.2.3.2 Pseudo Code

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

6.12.2.3 Show a Single Instance – Both References

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

6.12.2.3.1 Command Form

```
show <CIM_JobDestinationJobs single instance>
```

6.12.2.3.2 CIM Requirements

See CIM_JobDestinationJobs in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.12.2.3.3 Behavior Requirements

6.12.2.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_ConcreteJob which is referenced by
   CIM_JobDestinationJobs
2) $instanceB contains the instance of CIM_JobQueue which is referenced by
   CIM_JobDestinationJobs.
```

6.12.2.3.3.2 Pseudo Code

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

6.13 CIM_JobQueue

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

Table 12 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 12 is for informational purposes only; in case of a conflict between Table 12 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 12.

Table 12 – Command Verb Requirements for CIM_JobQueue

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 6.13.2.
start	Not supported	
stop	Not supported	

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

6.13.1 Ordering of Results

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

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

6.13.2 Show

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

The `show` verb is used to display information about the CLP Service's Job Queue.

6.13.2.1 Show a Single Instance

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

6.13.2.1.1 Command Form

```
show <CIM_JobQueue single instance>
```

6.13.2.1.2 CIM Requirements

See CIM_JobQueue in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.13.2.1.3 Behavior Requirements

6.13.2.1.3.1 Preconditions

```
#all is true if the all option was specified with the command; otherwise, #all is
false
$instance=<CIM_JobQueue Single Instance>
#propertylist[] = null;
if ( false == #all) {
   #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstance($instance.getObjectPath(), #propertylist[]);
```

6.13.2.2 Show Multiple Instances

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

6.13.2.2.1 Command Form

```
show <CIM_JobQueue single instance>
```

6.13.2.2.2 CIM Requirements

See CIM_JobQueue in the "CIM Elements" section of the *CLP Service Profile* for the list of mandatory properties.

6.13.2.2.3 Behavior Requirements

6.13.2.2.3.1 Preconditions

```
1) $containerInstance contains the instance of CIM_ComputerSystem for which we are
   displaying scoped CIM_JobQueue instances.
```

```
2) #all is true if the all option was specified with the command; otherwise, #all is
   false
```

6.13.2.2.3.2 Pseudo Code

```
#propertylist[] = null;
if (false == #all)  {
   #propertylist[] = {//all mandatory non-key properties }
}
&smShowInstances ( “CIM_JobQueue”, “CIM_HostedJobDestination",
   $containerInstance.getObjectPath(), #propertylist[] );
   &smEnd;
```

6.14 CIM_OwningJobElement

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

Table 13 lists each SM CLP verb, the required level of support for the verb in conjunction with instances of the target class, and, when appropriate, a cross-reference to the section detailing the mapping for the verb and target. Table 13 is for informational purposes only; in case of a conflict between Table 13 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 13.

Table 13 – Command Verb Requirements for CIM_OwningJobElement

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 6.14.2.
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`.

6.14.1 Ordering of Results

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

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

6.14.2 Show

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

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

6.14.2.1 Show a Single Instance – CIM_ConcreteJob Reference

This command form is for the show verb applied to a single instance. This command form corresponds to the show command issued against CIM_OwningJobElement where the reference specified is to an instance of CIM_ConcreteJob. A single instance of CIM_ProtocolService is associated with each instance of a CIM_ConcreteJob Therefore, a single instance will be returned.

6.14.2.1.1 Command Form

```
show <CIM_OwningJobElement single instance>
```

6.14.2.1.2 CIM Requirements

See CIM_OwningJobElement in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.14.2.1.3 Behavior Requirements

6.14.2.1.3.1 Preconditions

```
1) $instance contains the instance of CIM_ConcreteJob which is referenced by
   CIM_OwningJobElement
```

6.14.2.1.3.2 Pseudo Code

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

6.14.2.2 Show Multiple Instances – CIM_ProtocolService Reference

This command form is for the show verb applied to a single instance. This command form corresponds to the show command issued against CIM_OwningJobElement where the reference specified is to an instance of CIM_ProtocolService. A single instance of CIM_ProtocolService is associated with multiple instances of CIM_ConcreteJob. Therefore, multiple instances may be returned.

6.14.2.2.1 Command Form

```
show <CIM_OwningJobElement multiple instances>
```

6.14.2.2.2 CIM Requirements

See CIM_OwningJobElement in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.14.2.2.3 Behavior Requirements

6.14.2.2.3.1 Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_OwningJobElement
```

6.14.2.2.3.2 Pseudo Code

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

6.14.2.3 Show a Single Instance – Both References

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

6.14.2.3.1 Command Form

```
show <CIM_OwningJobElement single instance>
```

6.14.2.3.2 CIM Requirements

See CIM_OwningJobElement in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.14.2.3.3 Behavior Requirements

6.14.2.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_ConcreteJob which is referenced by
   CIM_OwningJobElement
2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
   CIM_OwningJobElement.
```

6.14.2.3.3.2 Pseudo Code

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

6.15 CIM_ProtocolService

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

Table 14 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 14 is for informational purposes only; in case of a conflict between Table 14 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 14.

Table 14 – Command Verb Requirements for CIM_ProtocolService

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	May	See 6.15.2.
set	May	See 6.15.3.
show	Shall	See 6.15.4.
start	May	See 6.15.5.
stop	May	See 6.15.6.

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

6.15.1 Ordering of Results

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

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

6.15.2 Reset

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

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

6.15.2.1 Reset a Single Instance

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

6.15.2.1.1 Command Form

```
reset <CIM_ProtocolService single instance>
```

6.15.2.1.2 CIM Requirements

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

6.15.2.1.3 Behavior Requirements

```
$instance=<CIM_ProtocolService Single Instance>
smResetRSC($instance.getObjectPath());
&smEnd;
```

6.15.3 Set

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

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

6.15.3.1 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 *CLP Service Profile*.

6.15.3.1.1 Command Form

```
set <CIM_ProtocolService Single Instance> <propertyname>=<propertyvalue>
```

6.15.3.1.2 CIM Requirements

See CIM_ProtocolService in the “CIM Elements” section of the *CLP Service Profile* for the list of modifiable properties.

6.15.3.1.3 Behavior Requirements

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

6.15.3.2 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 *CLP Service Profile*.

6.15.3.2.1 Command Form

```
set <CIM_ProtocolService Single Instance> <propertyname1>=<propertyvalue1>
<propertyname n>=<propertyvalue n>
```

6.15.3.2.2 CIM Requirements

See CIM_ProtocolService in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.15.3.2.3 Behavior Requirements

```
$instance = <CIM_ProtocolService Single Instance>
#propertyNames[] = {<propertyname>};

for #i < n
{
   #propertyNames[#i] = <propertname#i>
   #propertyValues[#i] = <propertyvalue#i>
}

&smSetInstance($instance, #propertyNames[], #propertyValues[]);
&smEnd;
```

6.15.4 Show

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

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

6.15.4.1 Show a Single Instance

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

6.15.4.1.1 Command Form

```
show <CIM_ProtocolService single instance>
```

6.15.4.1.2 CIM Requirements

See CIM_ProtocolService in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.15.4.1.3 Behavior Requirements

6.15.4.1.3.1 Preconditions

```
#all is true if the all option was specified with the command; otherwise, #all is
false
$instance=<CIM_ProtocolService Single Instance>
```

6.15.4.1.3.2 Pseudo Code

```
#propertylist[] = null;
if (false == #all) {
#propertylist[] = { //all mandatory non-key properties };
}

&smShowInstance($instance.getObjectPath(), #propertylist[]);
&smEnd;
```

6.15.4.2 Show Multiple Instances

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

6.15.4.2.1 Command Form

```
show <CIM_ProtocolService multiple instances>
```

6.15.4.2.2 CIM Requirements

See CIM_ProtocolService in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.15.4.2.3 Behavior Requirements

6.15.4.2.3.1 Preconditions

```
1) $containerInstance contains the instance of CIM_ComputerSystem for which we are
   displaying scoped instances of the CIM_ProtocolService. The CLP Service Profile
   requires that the CIM_ProtocolService instance be associated with its scoping
   system via an instance of the CIM_HostedService association.
2) #all is true if the all option was specified with the command; otherwise, #all is
   false
```

6.15.4.2.3.2 Pseudo Code

```
#propertylist[] = null;
if (false == #all) {
   #propertylist[] = { //all mandatory non-key properties };
}

&smShowInstances ( “CIM_ProtocolService”, “CIM_HostedService",
$containerInstance.getObjectPath(), #propertylist[] );
&smEnd;
```

6.15.5 Start

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

The `start` verb is used to enable the CIM_ProtocolService.

6.15.5.1 Start a Single Instance

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

6.15.5.1.1 Command Form

```
start <CIM_ProtocolService single instance>
```

6.15.5.1.2 CIM Requirements

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

6.15.5.1.3 Behavior Requirements

```
$instance=<CIM_ProtocolService Single Instance>
smStartRSC($instance.getObjectPath());
&smEnd;
```

6.15.6 Stop

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

The `stop` verb is used to disable the CIM_ProtocolService.

6.15.6.1 Stop a Single Instance

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

6.15.6.1.1 Command Form

```
stop <CIM_ProtocolService single instance>
```

6.15.6.1.2 CIM Requirements

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

6.15.6.1.3 Behavior Requirements

```
$instance=<CIM_ProtocolService Single Instance>
smStopRSC($instance.getObjectPath());
&smEnd;
```

6.16 CIM_ProvidesEndpoint

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

Table 15 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 15 is for informational purposes only; in case of a conflict between Table 15 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 15.

Table 15 – Command Verb Requirements for CIM_ProvidesEndpoint

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 6.16.2.
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`.

6.16.1 Ordering of Results

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

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

6.16.2 Show

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

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

6.16.2.1 Show a Single Instance – CIM_CLPProtocolEndpoint Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_ProvidesEndpoint where the reference specified is to an instance of CIM_CLPProtocolEndpoint. A single instance of CIM_ProtocolService is associated with each instance of a CIM_CLPProtocolEndpoint. Therefore, a single instance will be returned.

6.16.2.1.1 Command Form

```
show <CIM_ProvidesEndpoint single instance>
```

6.16.2.1.2 CIM Requirements

See CIM_ProvidesEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.16.2.1.3 Behavior Requirements

6.16.2.1.3.1 Preconditions

```
1) $instance contains the instance of CIM_CLPProtocolEndpoint which is referenced by
   CIM_ProvidesEndpoint
```

6.16.2.1.3.2 Pseudo Code

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

6.16.2.2 Show Multiple Instances – CIM_ProtocolService Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_ProvidesEndpoint where the reference specified is to an instance of CIM_ProtocolService. A single instance of CIM_ProtocolService is associated with multiple instances of a CIM_CLPProtocolEndpoint. Therefore, multiple instances may be returned.

6.16.2.2.1 Command Form

```
show <CIM_ProvidesEndpoint multiple instances>
```

6.16.2.2.2 CIM Requirements

See CIM_ProvidesEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.16.2.2.3 Behavior Requirements

6.16.2.2.3.1 Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_ProvidesEndpoint
```

6.16.2.2.3.2 Pseudo Code

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

6.16.2.3 Show a Single Instance – Both References

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

6.16.2.3.1 Command Form

```
show <CIM_ProvidesEndpoint single instance>
```

6.16.2.3.2 CIM Requirements

See CIM_ProvidesEndpoint in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.16.2.3.3 Behavior Requirements

6.16.2.3.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_CLPProtocolEndpoint which is referenced by
   CIM_ProvidesEndpoint
2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
   CIM_ProvidesEndpoint.
```

6.16.2.3.3.2 Pseudo Code

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

6.17 CIM_ServiceAffectsElement

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

Table 16 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 16 is for informational purposes only; in case of a conflict between Table 16 and requirements detailed in the following sections, the text detailed in the following sections supersedes the information in Table 16.

Table 16 – Command Verb Requirements for CIM_ServiceAffectsElement

Command Verb	Requirement	Comments
create	Not supported	
delete	Not supported	
dump	Not supported	
load	Not supported	
reset	Not supported	
set	Not supported	
show	Shall	See 6.17.2.
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`.

6.17.1 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.

6.17.2 Show

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

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

6.17.2.1 Show Multiple Instances – CIM_ProtocolService Reference

This command form is for the show verb applied to a single instance. This command form corresponds to the show command issued against CIM_ServiceAffectsElement where the reference specified is to an instance of CIM_ProtocolService. Multiple instances of CIM_ServiceAffectsElement reference an instance of CIM_ProtocolService.

6.17.2.1.1 Command Form

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

6.17.2.1.2 CIM Requirements

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

6.17.2.1.3 Behavior Requirements

6.17.2.1.3.1 Preconditions

```
1) $instance contains the instance of CIM_ProtocolService which is referenced by
   CIM_ServiceAffectsElement
```

6.17.2.1.3.2 Pseudo Code

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

6.17.2.2 Show a Single Instance – CIM_JobQueue Reference

This command form is for the show verb applied to a single instance. This command form corresponds to the show command issued against CIM_ServiceAffectsElement where the reference specified is to an instance of CIM_JobQueue. A single instance of CIM_ProtocolService is associated with each instance of a CIM_JobQueue. Therefore, a single instance will be returned.

6.17.2.2.1 Command Form

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

6.17.2.2.1.1 CIM Requirements

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

6.17.2.2.2 Behavior Requirements

6.17.2.2.2.1 Preconditions

```
1) $instance contains the instance of CIM_JobQueue which is referenced by
   CIM_ServiceAffectsElement
```

6.17.2.2.2.2 Pseudo Code

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

6.17.2.3 Show a Single Instance – CIM_AdminDomain Reference

This command form is for the `show` verb applied to a single instance. This command form corresponds to the `show` command issued against CIM_ServiceAffectsElement where the reference specified is to an instance of CIM_AdminDomain. A single instance of CIM_ProtocolService is associated with each instance of a CIM_AdminDomain. Therefore, a single instance will be returned.

6.17.2.3.1 Command Form

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

6.17.2.3.2 CIM Requirements

See CIM_AffectsElement in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.17.2.3.3 Behavior Requirements

6.17.2.3.3.1 Preconditions

```
$instance contains the instance of CIM_AdminDomain which is referenced by
   CIM_ServiceAffectsElement
```

6.17.2.3.3.2 Pseudo Code

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

6.17.2.4 Show a Single Instance – 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_ServiceAffectsElement where both references are specified and therefore the desired instance is unambiguously identified.

6.17.2.4.1 Command Form

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

6.17.2.4.2 CIM Requirements

See CIM_AffectsElement in the “CIM Elements” section of the *CLP Service Profile* for the list of mandatory properties.

6.17.2.4.3 Behavior Requirements

6.17.2.4.3.1 Preconditions

```
1) $instanceA contains the instance of CIM_JobQueue or CIM_AdminDomain that is
   referenced by CIM_ServiceAffectsElement
2) $instanceB contains the instance of CIM_ProtocolService which is referenced by
   CIM_ServiceAffectsElement.
```

6.17.2.4.3.2 Pseudo Code

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

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.0.0	7/29/09		DMTF Standard Release