Profile Error: This profile contains 3 errors (search for 'Error:')

Example Fan Profile

Document Type: Specification

Document Status: Work in Progress

Document Language: en-US

Copyright notice

CONTENTS

Figures

Tables

Foreword

This document was prepared by the Physical Platform Profiles Working Group and Server Management Working Group.

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

Design Note: This document contains design notes (like this one), that provide information about the way the document is written, or to demonstrate certain things. Such design notes would not appear in a released version of this document.

Design Note: This document represents DSP1013 (Fan Profile) version 1.0.1 plus some additions as a machine readable profile in MRP 1.1 format. Since machine readable profiles need to be compliant to DSP1001 1.1, this document utilizes the newly introduced concepts, such as adaptations, features and collaboration diagrams. Relative to DSP1013 1.0, this machine readable profile adds the following, in order to demonstrate its use:

- The use of standard messages defined in DSP8016 and DSP8007 as error messages, in the RequestStateChange() method of the Fan adaptation.

- The use of standard metrics defined in an assumed metric registry DSPsamr, in two variants: (1) The definition of a metric 'Metric1' directly on the Fan adaptation, demonstrating the most simple approach to defining metrics; (2) The definition of a metric 'Metric2' on a metric definition represented by the FanMetricDefinition adaptation, demonstrating the more flexible approach of defining a metric using a metric definition. In both variants, the metrics are represented using the base metric model defined in DSP1053 (Base Metrics Profile).

- The use of alert indications, demonstrating the most simple approach to defining alert indications as defined in DSP1001 1.1 and DSP1054 1.2.

Acknowledgements

DMTF acknowledges the following individuals for their contributions to this document:

- Jon Hass, Dell (editor)
- Khachatur Papanyan, Dell (editor)
- Jeff Hilland, HP (editor)
- Jim Davis, WBEM Solutions (editor)
- Enoch Suen, Dell
- Christina Shaw, HP
- Aaron Merkin, IBM
- Perry Vincent, Intel
- John Leung, Intel
- John Ackerley, Sun Microsystems

Document conventions

Any text in this document is in normal text font, with the following exceptions:

- References to clause names use normal text font; if they consist of more than one word, the clause name is quoted using double quotes, such as in "CIM elements".

- Important terms that are used for the first time are marked in *italics* .

- The usage of terms link to the term definition defined in the "Terms and definitions" clause, enabling easy navigation to the term definition.

- ABNF rules are in `monospaced font` .

Format definitions in this document are specified using ABNF (see RFC5234), with the following deviations:

- Literal strings are to be interpreted as case-sensitive Unicode characters, as opposed to the definition in RFC5234 that interprets literal strings as case-insensitve US-ASCII characters.

Example Fan Profile

1 Scope

The Fan Profile extends the management capabilities of referencing profiles by adding the capability to represent fans for manageability and describe redundant fans . The fan as a logical device is modeled as referencing the fan physical package for physical asset information, a sensor for sensor reading information, and the profile registration for the schema implementation version information.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For undated and unversioned references, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

DMTF DSP0004, *CIM Infrastructure Specification 2.5*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

DMTF DSP0223, *Generic Operations 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

DMTF XMP1009, *Example Sensors Profile (sample profile in DSP2023) 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip

DMTF XMP1011, *Example Physical Asset Profile (sample profile in DSP2023) 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip

DMTF XMP1033, *Example Profile Registration Profile (sample profile in DSP2023) 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip

DMTF DSP1053, *Base Metric Profile 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1053_1.1.pdf

DMTF DSP1054, *Indications Profile 1.2*,
http://www.dmtf.org/standards/published_documents/DSP1054_1.2.pdf

DMTF DSP8016, *WBEM Operations Message Registry 1.0*,
http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.0.xml

DMTF DSP8007, *Platform Message Registry 1.0*,
http://schemas.dmtf.org/wbem/messageregistry/1/dsp8007_1.0.xml

DMTF DSPsamr, *Sample Metric Registry 1.0*,
dspsamr_1.0.xml

IETF RFC5234, *Augmented BNF for Syntax Specifications: ABNF, 2008-01*,
http://tools.ietf.org/html/rfc5234

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

3 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

3.1 General

The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"), "may", "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part2 , Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part2 , Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning in this document.

The terms "clause", "subclause", "paragraph", "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part2 , Clause 5.

The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part2 , Clause 3. In this document, clauses, subclauses or annexes indicated with "(informative)" as well as notes and examples do not contain normative content.

The terms defined in DSP0004, DSP0223, and DSP1001 apply to this document.

The following additional terms are defined in this document.

3.2
fan
A device that provides thermal cooling by air flow to system elements.

3.3
redundant fan
A fan that is participating in a redundant set of fans.

3.4
spare fan
A fan that is not currently used, but is available for use in situations where currently used fans are no longer used for some reason.

4 Symbols and abbreviated terms

This clause defines the symbols and abbreviations used in this document.

The abbreviations defined in DSP0004, DSP0223, and DSP1001 apply to this document.

This document does not define any additional abbreviations.

5 Synopsis

Profile name: Example Fan

Version: 1.0.2

Organization: DMTF

Abstract: No

Profile type: Component

Schema: DMTF CIM 2.19

Central class adaptation: Fan

Scoping class adaptation: ComputerSystem

Scoping path: SystemDevice

The Example Fan profile extends the management capabilities of referencing profiles by adding the capability to represent fans as logical devices, to monitor and control the state and speed of fans , to represent the relationship to elements cooled by fans , and to represent redundant fans as a redundancy group. Optionally, the Sensors profile can be implemented for fan speed sensors (SpeedSensors profile reference). Optionally, the Physical Asset profile can be implemented for fans (PhysicalAsset profile reference).

Table 1 identifies the profile references defined in this profile.

Table 1 – Profile references

Profile reference name	Profile name	Organi-zation	Version	Relation-ship	Description
PhysicalAsset	Example Physical Asset	DMTF	1.0	Optional	Used to represent the physical packaging of fans.
SpeedSensors	Example Sensors	DMTF	1.0	Conditional	Used to represent fan speed sensors . Condition: The FanSpeedSensor feature is implemented.
PRP	Profile Registration	DMTF	1.0	Mandatory	Used to represent the implementation of this profile.
Indications	Indications	DMTF	1.2	Mandatory	Used for the indications defined by this profile.
BaseMetric	Base Metric	DMTF	1.1	Mandatory	Used for the metrics defined by this profile.

Table 2 identifies the message registry references defined in this profile.

Table 2 – Message registry references

Registry reference name	Registry ID	Organization	Version	Description
WBEMOperations	DSP8016	DMTF	1.0	
Platform	DSP8007	DMTF	1.0	

Table 3 identifies the metric registry references defined in this profile.

Table 3 – Metric registry references

Registry reference name	Registry ID	Organization	Version	Description
Sample	DSPsamr	DMTF	1.0	

Table 4 identifies the features defined in this profile.

Table 4 – Features

Feature	Requirement	Description
FanRedundancyByBalancing	Optional	See 7.1.1.
FanRedundancyBySparing	Optional	See 7.1.2.
FanSpeedSensor	Conditional	See 7.1.3.
PhysicalAssetDescription	Optional	See 7.1.4.
PartialCooling	Conditional	See 7.1.5.
FanCapabilities	Optional	See 7.1.6.
FanElementNameModification	Optional	See 7.1.7.
FanStateManagement	Optional	See 7.1.8.
SettingFanSpeed	Optional	See 7.1.9.
FanIndications	Optional	See 7.1.10.

Table 5 identifies the class adaptations defined in this profile.

Table 5 – Adaptations

Adaptation	Elements	Requirement	Description
Instantiated, embedded and abstract adaptations			
ComputerSystem	CIM_ComputerSystem	Mandatory	See 7.2.2.
SystemDevice	CIM_SystemDevice	Mandatory	See 7.2.3.
FanMetricDefinition	CIM_BaseMetricDefinition	Optional	See 7.2.4.
Fan	CIM_Fan	Mandatory	See 7.2.5.
FanCapabilities	CIM_EnabledLogicalElementCapabilities	Conditional	See 7.2.6.
ElementCapabilities	CIM_ElementCapabilities	Conditional	See 7.2.7.
CooledElement	CIM_ManagedSystemElement	Conditional	See 7.2.8.
AssociatedCooling	CIM_AssociatedCooling	Conditional	See 7.2.9.
FanRedundancySet	CIM_RedundancySet	Conditional	See 7.2.10.
OwningCollectionElement	CIM_OwningCollectionElement	Conditional	See 7.2.11.
HostedRedundancySet	CIM_HostedCollection	Conditional	See 7.2.12.
MemberOfRedundancySet	CIM_MemberOfCollection	Conditional	See 7.2.13.
IsSpare	CIM_IsSpare	Conditional	See 7.2.14.
NumericFanSpeedSensor	CIM_NumericSensor	Conditional	See 7.2.15.
DiscreteFanSpeedSensor	CIM_Sensor	Conditional	See 7.2.16.
FanSpeedAlertIndicationFilter	CIM_IndicationFilter	Conditional	See 7.2.21.
FanAddedLifecycleIndicationFilter	CIM_IndicationFilter	Conditional	See 7.2.22.
Indications and exceptions			
FanAddedIndication	CIM_InstCreation	Conditional	See 7.2.17.
FanRemovedIndication	CIM_InstDeletion	Conditional	See 7.2.18.
FanHealthIndication	CIM_AlertIndication	Conditional	See 7.2.19.
FanRedundancyIndication	CIM_AlertIndication	Conditional	See 7.2.20.

Table 6 identifies the use cases and state descriptions defined in this profile.

Table 6 – Use cases and state descriptions

Name	Description
State description: ObjectDiagram	See 8.1.
Use case: SetFanSpeed	See 8.2.
Use case: ResetFan	See 8.3.
Use case: GetFanRedundancyStatus	See 8.4.
Use case: FindSpareFan	See 8.5.
Use case: ShowFanSensorInfo	See 8.6.
Use case: FindCooledElements	See 8.7.
Use case: DetermineElementNameModifiability	See 8.8.

6 Description
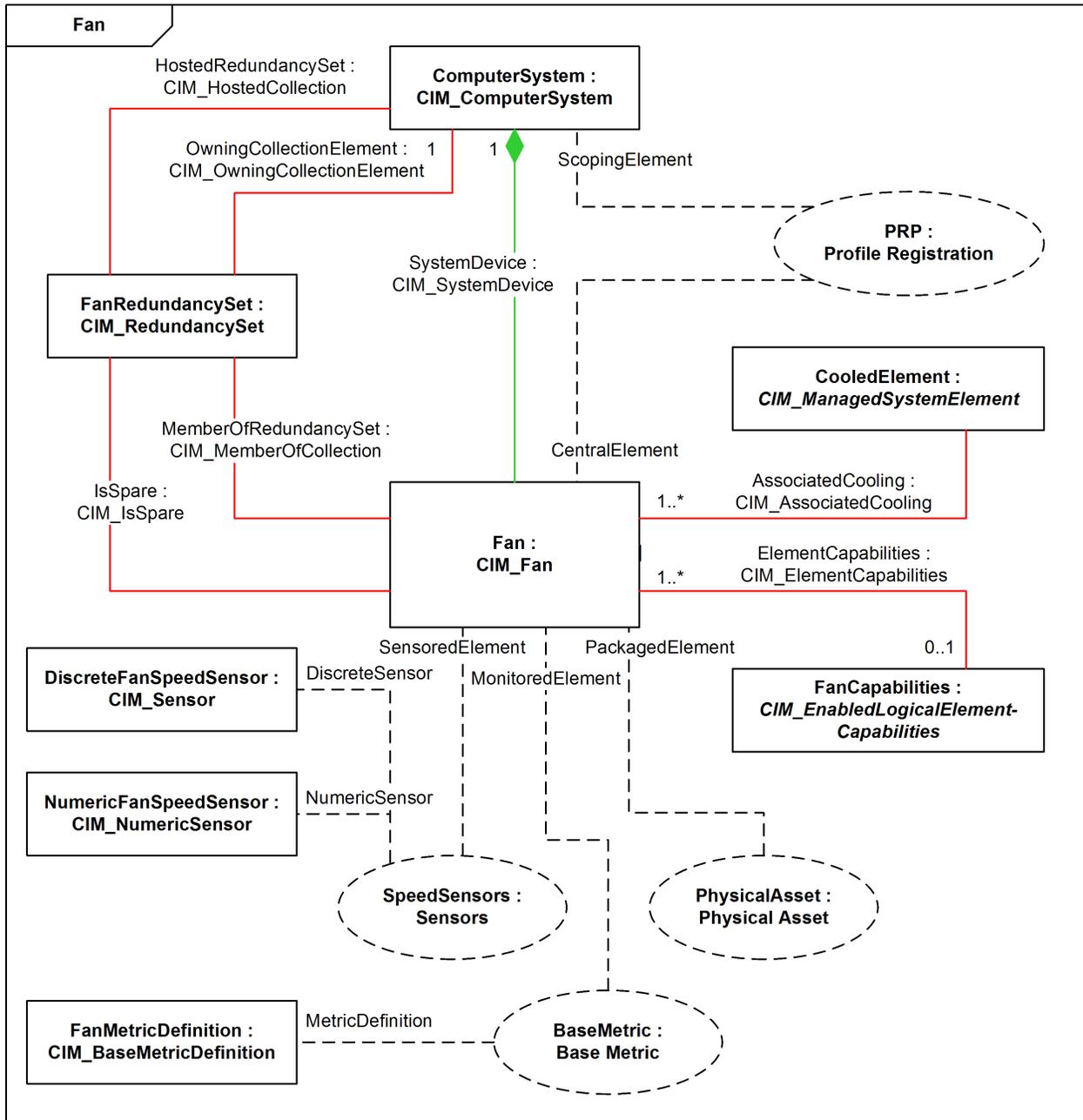
Figure 1 – DMTF collaboration structure diagram

The logical aspect of fans in the managed environment is represented by instances of the Fan adaptation. The system hosting the fans is represented by an associated ComputerSystem instance.

The capability to support the representation of fan redundancy can be added by implementing the FanRedundancyBySparing or FanRedundancyByBalancing features, which are mutually exclusive for a given fan. If one of these features is implemented for a fan, each redundancy group this fan is a member of is represented by an associated FanRedundancySet instance.

The capability to support the sensoring of the fan speed can be added by implementing the FanSpeedSensor feature for a fan, using the SpeedSensors profile.

The capability to expose physical asset information for a fan can be added by implementing the PhysicalAssetDescription feature for the fan, using the PhysicalAsset profile.

The capability to provide cooling only to specific elements of the system instead of the whole system can be added by implementing the PartialCooling feature. If it is implemented for a fan, the system element to which the fan provides cooling is represented by an associated CooledElement instance. If it is not implemented for a fan, the fan provides cooling to the entire system that hosts the fan.

The capability to expose the capabilities of a fan can be added by implementing the FanCapabilities feature. If it is implemented for a fan, its capabilities are represented by an associated FanCapabilities instance. Note that FanCapabilities instances can be shared between multiple Fan instances.

Conformance of an implementation to this profile is represented through the PRP profile.

7 Implementation

7.1 Features

7.1.1 Feature: FanRedundancyByBalancing

The implementation of this feature for a fan provides the ability to represent that a fan is redundant within a redundancy group of fans, such that all fans in the group run at the same time, balancing the cooling load between them. If one of these fan fails, the others remain running to provide cooling.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A Fan instance.

  ```
  derive: self.MemberOfRedundancySet::Collection->size() > 0 and
  self.IsSpare.Antecedent->size() = 0
  ```

Otherwise, it can be concluded that the feature is not available.

7.1.2 Feature: FanRedundancyBySparing

The implementation of this feature for a fan provides the ability to represent that a fan is redundant within a redundancy group of fans, such that some fans in the group run at the same time, balancing the cooling load between them, and some others do not normally run and act as spare fans . If one of the normally running fans fails, the other normally running fans remain running to provide cooling; and additional spare fans may be started to accomodate for the cooling capacity of the failed fan(s).

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A Fan instance.

```
derive: self.MemberOfRedundancySet::Collection->size() > 0 and
self.IsSpare.Antecedent->size() > 0
```

Otherwise, it can be concluded that the feature is not available.

7.1.3 Feature: FanSpeedSensor

This feature provides the ability to expose the speed of fans that have a speed sensor .

The requirement level for this feature is conditional, with the following condition:

> The feature should be implemented for fans that have a speed sensor .

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A Fan instance.

```
derive: self.SpeedSensors::NumericSensor->size() > 0 or
self.SpeedSensors::DiscreteSensor->size() > 0
```

Otherwise, it can be concluded that the feature is not available.

7.1.4 Feature: PhysicalAssetDescription

This feature provides support for describing physical asset information of a fan.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of an implementation of this profile.

It can be concluded that the feature is available if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

  OCL context: A RegisteredProfile instance for this profile.

```
derive: self.mrpIsReferencedProfileImplemented('PhysicalAsset')
```

Otherwise, it can be concluded that the feature is not available.

7.1.5 Feature: PartialCooling

The implementation of this feature for a fan allows representing that the fan provides cooling to a subset of the elements in a system, instead of to the entire system.

The requirement level for this feature is conditional, with the following condition:

> The feature shall be implemented for a fan if the fan provides cooling to a subset of the elements in the system. The feature shall not be implemented for a fan if the fan provides cooling to the entire system.

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

- The following OCL derivation constraint evaluates to a Boolean value of True.

OCL context: A Fan instance.

```
derive: self.AssociatedCooling::Antecedent->size() > 0
```

Otherwise, it can be concluded that the feature is not available.

7.1.6 Feature: FanCapabilities

This feature allows a fan to expose its capabilities through an FanCapabilities instance.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

  • The following OCL derivation constraint evaluates to a Boolean value of True.

    OCL context: A Fan instance.

```
derive: self.ElementCapabilities->size() = 1
```

    Explanation:

         One instance exists of ElementCapabilities that is associated to the Fan instance.

Otherwise, it can be concluded that the feature is not available.

7.1.7 Feature: FanElementNameModification

This feature provides support for client modification of the CIM_Fan.ElementName property of a fan.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

  • The following OCL derivation constraint evaluates to a Boolean value of True.

    OCL context: A Fan instance.

```
derive: self.ElementCapabilities::Capabilities.ElementNameEditSupported = true
```

Otherwise, it can be concluded that the feature is not available.

7.1.8 Feature: FanStateManagement

This feature provides support for client management of the state of a fan.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of Fan instances.

It can be concluded that the feature is available for a Fan instance if:

  • The following OCL derivation constraint evaluates to a Boolean value of True.

    OCL context: A Fan instance.

```
derive: self.ElementCapabilities::Capabilities.RequestedStatesSupported->size()
> 0
```

Otherwise, it can be concluded that the feature is not available.

7.1.9 Feature: SettingFanSpeed

This feature provides support for setting the speed of a fan, via the CIM_Fan.SetSpeed() method.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of Fan instances.

Availability of this feature cannot be discovered by clients (other than trying the functionality provided by the feature).

7.1.10 Feature: FanIndications

Design Note: This feature has been defined to demonstrate the grouping of indication implementation decisions under one point of decision; this features is not part of DSP1013 1.0. Note, a profile is free to define more granular features for indications, or to define indications without usage of any feature for grouping purposes.

This feature provides support for indications related to a fan.

The requirement level for this feature is optional.

This feature can be made available to clients at the granularity of an implementation of this profile.

Availability of this feature cannot be discovered by clients (other than trying the functionality provided by the feature).

7.2 Adaptations

7.2.1 Conventions

This profile defines operation requirements based on DSP0223.

For adaptations of ordinary classes and of associations, the requirements for operations are defined in adaptation-specific subclauses of subclause 7.2.

For association traversal operation requirements that are specified only in the elements table of an adaptation (i.e., without operation-specific subclauses), the names of the association adaptations to be traversed are listed in the elements table.

The default initialization requirement level for property requirements is optional.

The default modification requirement level for property requirements is optional.

This profile repeats the effective values of certain Boolean qualifiers as part of property, method parameter, or method return value requirements. The following convention is established: If the name of a qualifier is listed, its effective value is True; if the qualifier name is not listed, its effective value is False. The convention is applied in the following cases:

- In: indicates that the parameter is an input parameter
- Out: indicates that the parameter is an output parameter
- Key: indicates that the property is a key (that is, its value is part of the instance path)

- Required: indicates that the element value shall be non-Null

- Null OK: indicates explicitly that the element value may be Null for mandatory, conditional or conditional exclusive properties. This information is not specified as a qualifier in the schema but as an indicator in the profile.

7.2.2 Adaptation: ComputerSystem: CIM_ComputerSystem

7.2.2.1 General

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is mandatory.

Table 7 identifies the element requirements for this adaptation.

Table 7 – ComputerSystem: Element requirements

Element	Requirement	Description
Operations		
Associators( ) for HostedRedundancySet and others	Mandatory	See 7.2.2.2.
AssociatorNames( ) for HostedRedundancySet and others	Mandatory	See 7.2.2.3.
References( ) for HostedRedundancySet and others	Mandatory	See 7.2.2.4.
ReferenceNames( ) for HostedRedundancySet and others	Mandatory	See 7.2.2.5.

7.2.2.2 Operation: Associators( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet

- OwningCollectionElement

- SystemDevice

7.2.2.3 Operation: AssociatorNames( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet

- OwningCollectionElement

- SystemDevice

7.2.2.4 Operation: References( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet
- OwningCollectionElement
- SystemDevice

7.2.2.5 Operation: ReferenceNames( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet
- OwningCollectionElement
- SystemDevice

7.2.3 Adaptation: SystemDevice: CIM_SystemDevice

7.2.3.1 General

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is mandatory.

Table 8 identifies the element requirements for this adaptation.

Table 8 – SystemDevice: Element requirements

Element	Requirement	Description
Properties		
GroupComponent	Mandatory	Key, see 7.2.3.2.
PartComponent	Mandatory	Key, see 7.2.3.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.3.2 Property: GroupComponent

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation ComputerSystem.
- The multiplicity of [1 .. 1] defined in the schema is not further constrained.

7.2.3.3 Property: PartComponent

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation Fan.

- The multiplicity of [0 .. *] defined in the schema is constrained to [1 .. *].

7.2.4 Adaptation: FanMetricDefinition: CIM_BaseMetricDefinition

Design Note: This adaptation has been defined to demonstrate the definition of metrics through a metric definition; it is not part of DSP1013 1.0.

This adaptation models metric definitions for fans.

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is optional.

Table 9 identifies the element requirements for this adaptation.

Table 9 – FanMetricDefinition: Element requirements

Element	Requirement	Description
Base adaptations		
BaseMetric::BaseMetricDefinition	Optional	See BaseMetric::BaseMetricDefinition.
Metrics		
Sample::Metric1	Optional	Sample metric #1

7.2.5 Adaptation: Fan: CIM_Fan

7.2.5.1 General

This adaptation models fans in the managed environment.

Design Note: This adaptation defines an additional base adaptation 'MonitoredElement' and a metric 'Metric2' to demonstrate the definition of metrics; this is not part of DSP1013 1.0.

Each fan in the managed environment should be represented by a Fan instance.

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is mandatory.

The implementation shall satisfy the following constraints for this adaptation:

- OCL constraint with context of a Fan instance:

```
inv:
( mrpIsFeatureSupported('FanCapabilities',self) or
  mrpIsFeatureSupported('FanElementNameModification',self) )
implies
        self.cIM_ElementCapabilities.Capabilities->size() = 1
```

Explanation:

> If the FanCapabilities feature or the FanElementNameModification feature are supported for a fan, then there shall be one FanCapabilities instance associated via ElementCapabilities to the Fan instance representing that fan.

- OCL constraint with context of a Fan instance:

```
inv:
self.SystemDevice.ComputerSystem->size() = 1
```

Explanation:

> There shall be one ComputerSystem instance associated via SystemDevice to the Fan instance representing that fan.

- OCL constraint with context of a Fan instance:

```
inv:
mrpIsFeatureSupported('FanRedundancyByBalancing',self) or
mrpIsFeatureSupported('FanRedundancyBySparing',self)
implies
    let rgfans : Set(Fan) =
        self.MemberOfCollection.Collection.MemberOfCollection.Member
        /* rgfans is the set of fans in the redundancy group of the
            current redundant fan (i.e. self) */
    in
        if mrpIsFeatureSupported('PartialCooling',self)
        then rgfans->forall( rgfan |
            rgfan.AssociatedCooling.Dependent =
                self.AssociatedCooling.Dependent)
        else rgfans->forall( rgfan |
            rgfan.SystemDevice.System =
                self.SystemDevice.System)
        endif
```

Explanation:

> If feature 'fan redundancy' is supported for a fan and feature 'partial cooling' is supported for the same fan, the CIM_Fan instances in the redundancy group of that fan shall be associated with the same CIM_ManagedSystemElement instance through CIM_AssociatedCooling associations.

> If feature 'fan redundancy' is supported for a fan and feature 'partial cooling' is not supported for the same fan, the CIM_Fan instances in the redundancy group of that fan shall be associated with the same CIM_ComputerSystem instance through CIM_SystemDevice associations.

Table 10 identifies the element requirements for this adaptation.

Table 10 – Fan: Element requirements

Element	Requirement	Description
Base adaptations		
SpeedSensors::SensoredElement	Optional	See SpeedSensors::SensoredElement.
PhysicalAsset::PackagedElement	Optional	See PhysicalAsset::PackagedElement.
BaseMetric::MonitoredElement	Optional	See BaseMetric::MonitoredElement.
Metrics		
Sample::Metric2	Optional	Sample metric #2
Properties		
SystemCreationClassName	Mandatory	Key, see schema definition.
SystemName	Mandatory	Key, see schema definition.
CreationClassName	Mandatory	Key, see schema definition.
DeviceID	Mandatory	Key, see schema definition.

Element	Requirement	Description
OperationalStatus	Mandatory	See schema definition.
HealthState	Mandatory	See schema definition.
VariableSpeed	Mandatory	See schema definition.
DesiredSpeed	Mandatory	See 7.2.5.2.
ActiveCooling	Mandatory	See 7.2.5.3.
EnabledState	Mandatory	See 7.2.5.4.
RequestedState	Mandatory	See 7.2.5.5.
ElementName	Mandatory	See 7.2.5.6.
Methods		
RequestStateChange( )	Conditional	See 7.2.5.7.
SetSpeed( )	Conditional	See 7.2.5.8.
Operations		
GetInstance( )	Mandatory	See 7.2.5.9.
EnumerateInstances( )	Mandatory	See DSP0223.
EnumerateInstanceNames( )	Mandatory	See DSP0223.
ModifyInstance( )	Conditional	See 7.2.5.10.
Associators( ) for AssociatedCooling and others	Mandatory	See 7.2.5.11.
AssociatorNames( ) for AssociatedCooling and others	Mandatory	See 7.2.5.12.
References( ) for AssociatedCooling and others	Mandatory	See 7.2.5.13.
ReferenceNames( ) for AssociatedCooling and others	Mandatory	See 7.2.5.14.

7.2.5.2 Property: DesiredSpeed

The presentation requirement level for this property is mandatory.

If setting the fan speed is supported, the meaning of the value 0 is that no change in fan speed has been requested.

If setting the fan speed is not supported, the value of this property is meaningless.

The implementation shall satisfy the following constraints for this property:

- OCL constraint with context of a Fan instance:

```
init:
mrpIsFeatureSupported('SettingFanSpeed',self)
implies
        self.DesiredSpeed = 0
```

Explanation:

If setting the fan speed is supported for the fan, the initial value of DesiredSpeed shall be 0.

- OCL constraint with context of a Fan instance:

```
inv:
not mrpIsFeatureSupported('SettingFanSpeed',self)
```

```
implies
        self.DesiredSpeed = 0
```

Explanation:

> If setting the fan speed is not supported for the fan, the value of DesiredSpeed shall be 0.

7.2.5.3 Property: ActiveCooling

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

> OCL constraint with context of a Fan instance:

```
inv:
self.ActiveCooling = True
```

Explanation:

> ActiveCooling shall match True.

7.2.5.4 Property: EnabledState

The presentation requirement level for this property is mandatory.

Table 11 describes the mapping between the EnabledState property values and the corresponding description of the state of the fan.

Table 11 – EnabledState Value Description

Value	Description	Extended Description
2	Enabled	The fan shall be turned on.
3	Disabled	The fan shall be turned off.
5	Not Applicable	The fan state is indeterminate, or fan state management is not supported.

The value of the EnabledState property may change as the result of a change to the fan's enabled state by a non-CIM implementation.

7.2.5.5 Property: RequestedState

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this property:

• OCL constraint with context of a Fan instance:

```
inv:
mrpIsFeatureSupported('FanStateManagement',self)
implies
        Set { 5 /* No Change */, 12 /* Not Applicable */ }->
                union( self.ElementCapabilities.Capabilities->
                asOrderedSet()->at(1).RequestedStatesSupported)->
                includes(self.RequestedState)
```

Explanation:

If feature 'FanStateManagement' is supported for a fan, the value of RequestedState shall be 5 (No Change), 12 (Not Applicable), or one of the values in the RequestedStatesSupported array of the associated CIM_EnabledLogicalElementCapabilities instance.

- OCL constraint with context of a Fan instance:

```
inv:
not mrpIsFeatureSupported('FanStateManagement',self)
implies
        self.RequestedState = 12 /* Not Applicable */
```

Explanation:

If feature 'fan state management' is not supported for a fan, the value of the RequestedState property shall be 12 (Not Applicable).

- OCL constraint with context of a Fan instance:

```
init:
mrpIsFeatureSupported('FanStateManagement',self)
implies
        self.RequestedState = 5 /* No Change */
```

Explanation:

If feature 'FanStateManagement' is supported for a fan, the initial value of RequestedState shall be 5 (No Change).

- OCL constraint with context of a Fan instance:

```
inv:
not mrpIsFeatureSupported('FanStateManagement',self)
implies
        self.RequestedState = 12 /* Not Applicable */
```

Explanation:

If feature 'FanStateManagement' is not supported for a fan, the value of the RequestedState property shall be 12 (Not Applicable).

- OCL constraint with context of a Fan instance:

```
inv:
not mrpIsFeatureSupported('FanStateManagement',self)
implies
        self.RequestedState = 12 /* Not Applicable */
```

Explanation:

If feature 'fan state management' is supported for a fan, the value of the RequestedState property shall be 12 (Not Applicable).

7.2.5.6 Property: ElementName

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a Fan instance:

```
inv:
self.ElementCapabilities.Capabilities.
    ElementNameEditSupported = True
implies
    self.ElementName.isModifiable()
```

Explanation:

The ElementName property shall be modifiable when the ElementNameEditSupported
property of the CIM_EnabledLogicalElementCapabilities instance that is associated with the
CIM_Fan instance has a value of True.

7.2.5.7 Method: RequestStateChange( )

The requirement level for this method is conditional, with the following condition:

The FanStateManagement feature is implemented.

Design Note: This method defines error reporting requirements based on standard messages. It
represents the error situations 'method not supported' and 'timeout' as errors, instead of as return values
as defined in DSP1013 1.0.

If feature 'fan state management' is supported, the method shall be implemented.

When the RequestStateChange() method does not complete successfully and the fan is in an
indeterminate state, the EnabledState property shall have a value of 5 (Not Applicable).

Invocation of the RequestStateChange() method changes the fan's state to the value specified in the
RequestedState parameter.

Invoking this method multiple times may result in earlier requests being overwritten or lost.

The implementation shall satisfy the following constraints for this method:

• OCL constraint with context of a Fan instance:

```
pre:
Set { 2 /* Enabled */, 3 /* Disabled */, 11 /* Reset */ }->
    includes(self.RequestStateChange.RequestedState)
```

Explanation:

The RequestedState parameter shall have one of the following values: 2 (Enabled), 3
(Disabled), 11 (Reset).

• OCL constraint with context of a Fan instance:

```
post:
self.RequestedState = self.RequestStateChange.RequestedState
```

Explanation:

After successful completion of the RequestStateChange() method, the RequestedState
property of the CIM_Fan instance for which the method was invoked, shall have the value
specified in the RequestedState parameter.

- OCL constraint with context of a Fan instance:

```
post:
self.IsSpare->size() > 0 and
    self.RequestStateChange.RequestedState = 3 /* Disabled */
implies
        self.IsSpare.SpareStatus = 3 /* Cold Standby */
```

Explanation:

After successful completion of the RequestStateChange() method on a CIM_Fan instance representing a spare fan with the RequestedState parameter set to 3 (Disabled), the SpareStatus property of the CIM_IsSpare association referencing this CIM_Fan instance shall have a value of 3 (Cold Standby).

Table 12 identifies the parameter and return value requirements of the method.

Table 12 – RequestStateChange( ): Parameter requirements

Parameter	Description
RequestedState	In, see 7.2.5.7.1.
Job	Out, see 7.2.5.7.2.
TimeoutPeriod	In, see 7.2.5.7.3.
ReturnValue	See 7.2.5.7.4.

Table 13 specifies the error reporting requirements of the RequestStateChange( ) method. These requirements apply in addition to those defined in DSP0223 for the operation that invokes this method.

Table 13 – RequestStateChange( ): Error reporting requirements

Reporting mechanism	Requirement level	Description
WBEMOperations::WIPG0208, Platform::PLAT9001	Mandatory	The requested state is not supported for the fan. Design Note: The messages are: • WIPG0208: Invalid input parameter value • PLAT9001(example): Requested state not supported for the element
WBEMOperations::WIPG0208, Platform::PLAT9002	Mandatory	A non-Null value for the Timeout parameter is not supported. Design Note: The messages are: • WIPG0208: Invalid input parameter value • PLAT9002(example): Timeout not supported for the method
WBEMOperations::WIPG0219	Mandatory	Method is not implemented. Note: This error situation and its reporting through this message is defined already in DSP0223. This error situation is listed in this table only because it was

Reporting mechanism	Requirement level	Description
		reported through a return value in earlier versions of this profile. Design Note: The messages are: • WIPG0219: Method not supported by class implementation
WBEMOperations::WIPG0227, Platform::PLAT9003	Mandatory	Fan cannot be disabled due to excessive temperature. The detail text of WIPG0227 should be omitted or should indicate that the next message details the error. Design Note: The messages are: • WIPG0227: Other failure • PLAT9003(example): Fan cannot be disabled due to excessive temperature
WBEMOperations::WIPG0227	Mandatory	Any other failure. As defined in WIPG0227, the failure shall be described in its detail text. Note: This error situation and its reporting through this message is defined already in DSP0223. This error situation is listed in this table only because it was reported through a return value in earlier versions of this profile. Design Note: The messages are: • WIPG0227: Other failure
CIM_ERR_SERVER_LIMITS_EXCEEDED	Mandatory	More element changes are under way than the configured limit of concurrent changes, or there is a resource shortage in the WBEM server. Design Note: This error situation demonstrates the possibility of mixing the usage of CIM status codes and messages. It is not recommended to define such a mixed usage in a single profile, but it may happen in merged profiles.

7.2.5.7.1 Parameter: RequestedState

For valid values, see method constraints.

7.2.5.7.2 Parameter: Job

A non-Null instance path is returned if a job was started. If no job was started, Null is returned.

The implementation shall satisfy the following constraint for this reference parameter:

> Referenced instances shall be of class adaptation ConcreteJob**Profile Error: A class adaptation "ConcreteJob" is referenced in a class adaptation link but is not defined or is defined more than once in this profile.**.

7.2.5.7.3 Parameter: TimeoutPeriod

Client-specified maximum amount of time the transition to a new state is supposed to take:

- 0 or Null – No maximum time is specified
- Non-Null – The value specifies the maximum time allowed

7.2.5.7.4 Return value

This method shall return one of the following return values:

Table 14 – RequestStateChange: Return values

Value	Description
0	The state change was successfully performed.
1	The method is not implemented.
2	An error has occurred.
4096	The request to change the state is being executed asynchronously, and the Job parameter references a ConcreteJob**Profile Error: A class adaptation "ConcreteJob" is referenced in a class adaptation link but is not defined or is defined more than once in this profile.** instance representing the request.

7.2.5.8 Method: SetSpeed( )

The requirement level for this method is conditional, with the following condition:

> The SettingFanSpeed feature is implemented.

If the feature is not supported, the method shall not be implemented or shall return a value of 1 (Not Supported).

The SetSpeed() method requests that the speed of the fan represented by CIM_Fan be set to the value specified in the method's DesiredSpeed input parameter.

The implementation shall satisfy the following constraints for this method:

- OCL constraint with context of a Fan instance:

```
body:
if (self.VariableSpeed = false or
    self.EnabledState = 3 /* Disabled */ )
then 1 /* Not Supported */
```

Explanation:

> When the CIM_Fan.VariableSpeed property has a value of FALSE or the CIM_Fan.EnabledState property has a value of 3 (Disabled), the CIM_Fan.SetSpeed() method shall return a value of 1 (Not Supported).

- OCL constraint with context of a Fan instance:

```
post:
self.DesiredSpeed = self.SetSpeed.Speed
```

Explanation:

> When the CIM_Fan.SetSpeed() method successfully executed, the DesiredSpeed property shall be the value of the Speed parameter of the SetSpeed() method.

Table 15 identifies the parameter and return value requirements of the method.

Table 15 – SetSpeed( ): Parameter requirements

Parameter	Description
DesiredSpeed	In, see 7.2.5.8.1.
ReturnValue	See 7.2.5.8.2.

7.2.5.8.1 Parameter: DesiredSpeed

The parameter shall be specified in a unit of RPMs (revolutions per minute)

7.2.5.8.2 Return value

This method shall return one of the following return values:

Table 16 – SetSpeed: Return values

Value	Description
0	The speed change was successfully performed.
1	The method is not implemented.
2	An error has occurred. Note, the meaning of this value differs from the definition in the CIM schema.

7.2.5.9 Operation: GetInstance( )

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

Table 17 specifies the error reporting requirements of the GetInstance( ) operation. These requirements apply in addition to those defined in DSP0223 for this operation.

Table 17 – GetInstance( ): Error reporting requirements

Reporting mechanism	Requirement level	Description
WBEMOperations::WIPG0209, Platform::PLAT1234	Mandatory	

7.2.5.10 Operation: ModifyInstance( )

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is conditional, with the following condition:

> The FanElementNameModification feature is implemented.

The implementation shall satisfy the following constraints for this operation:

- OCL constraint with context of a Fan instance:

```
post:
if self.ElementCapabilities.Capabilities.
    ElementNameEditSupported = True
then self.ElementName =
    self.ModifyInstance.ModifiedInstance.ElementName
else self.ElementName =
    self.ElementName
```

Explanation:

> When the ElementNameEditSupported property of the
> CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_Fan
> instance has a value of True, the implementation shall allow the ModifyInstance operation
> to change the value of the ElementName property of the CIM_Fan instance. When the
> ElementNameEditSupported property of the CIM_EnabledLogicalElementCapabilities
> instance has a value of FALSE, the implementation shall not allow the ModifyInstance
> operation to change the value of the ElementName property of the CIM_Fan instance.

• OCL constraint with context of a Fan instance:

```
post:
self.EjlementCapabilities.Capabilities.
    ElementNameEditSupported = True
implies
    self.ElementName.size() <=
        self.cIM_ElementCapabilities.Capabilities.MaxElementNameLen
```

Explanation:

> When the ElementNameEditSupported property of the
> CIM_EnabledLogicalElementCapabilities instance that is associated with the CIM_Fan
> instance has a value of True, the implementation shall allow the ModifyInstance operation
> to change the value of the ElementName property of the CIM_Fan instance. The
> ModifyInstance operation shall enforce the length restriction specified in the
> MaxElementNameLen property of the CIM_EnabledLogicalElementCapabilities instance.

7.2.5.11 Operation: Associators( ) for AssociatedCooling and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- AssociatedCooling
- ElementCapabilities
- IsSpare
- MemberOfRedundancySet
- SystemDevice

7.2.5.12 Operation: AssociatorNames( ) for AssociatedCooling and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- AssociatedCooling
- ElementCapabilities
- IsSpare
- MemberOfRedundancySet

    • SystemDevice

7.2.5.13   Operation: References( ) for AssociatedCooling and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

    • AssociatedCooling

    • ElementCapabilities

    • IsSpare

    • MemberOfRedundancySet

    • SystemDevice

7.2.5.14   Operation: ReferenceNames( ) for AssociatedCooling and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

    • AssociatedCooling

    • ElementCapabilities

    • IsSpare

    • MemberOfRedundancySet

    • SystemDevice

7.2.6   Adaptation: FanCapabilities: CIM_EnabledLogicalElementCapabilities

7.2.6.1   General

This adaptation models the capabilities of fans modeled with Fan .

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is conditional, with the following condition:

      The FanCapabilities feature is implemented.

Table 18 identifies the element requirements for this adaptation.

Table 18 – FanCapabilities: Element requirements

Element	Requirement	Description
Properties		
InstanceID	Mandatory	Key, see schema definition.
RequestedStatesSupported	Mandatory	See 7.2.6.2.
ElementNameEditSupported	Mandatory	See 7.2.6.3.
MaxElementNameLen	Conditional	See 7.2.6.4.

Element	Requirement	Description
Operations		
GetInstance( )	Mandatory	See DSP0223.
EnumerateInstances( )	Mandatory	See DSP0223.
EnumerateInstanceNames( )	Mandatory	See DSP0223.
Associators( ) for ElementCapabilities	Mandatory	See DSP0223.
AssociatorNames( ) for ElementCapabilities	Mandatory	See DSP0223.
References( ) for ElementCapabilities	Mandatory	See DSP0223.
ReferenceNames( ) for ElementCapabilities	Mandatory	See DSP0223.

7.2.6.2 Property: RequestedStatesSupported

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a FanCapabilities instance:

```
inv:
if mrpIsFeatureSupported('fan ElementName modification',
    self.cIM_ElementCapabilities.Element)
then self.RequestedStatesSupported =
    Set { 2 /* Enabled */, 3 /* Disabled */, 11 /* Reset */ } )
else self.RequestedStatesSupported->isEmpty() )
```

Explanation:

If feature 'fan ElementName modification' is supported, the RequestedStatesSupported
array property shall contain any combination of the values: 2 (Enabled), 3 (Disabled), 11
(Reset). If feature 'fan ElementName modification' is not supported, the
RequestedStatesSupported property shall be an empty array.

7.2.6.3 Property: ElementNameEditSupported

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a FanCapabilities instance:

```
inv:
mrpIsFeatureSupported('fan ElementName modification',
    self.cIM_ElementCapabilities.Element)
implies
        self.ElementNameEditSupported = True
```

Explanation:

If client modification of the CIM_Fan.ElementName property is supported, the
ElementNameEditSupported property shall have a value of True.

7.2.6.4 Property: MaxElementNameLen

The presentation requirement level for this property is conditional, with the following condition:

The FanElementNameModification feature is implemented.

7.2.7   Adaptation: ElementCapabilities: CIM_ElementCapabilities

7.2.7.1   General

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> The FanCapabilities feature is implemented.

Table 19 identifies the element requirements for this adaptation.

Table 19 – ElementCapabilities: Element requirements

Element	Requirement	Description
Properties		
ManagedElement	Mandatory	Key, see 7.2.7.2.
Capabilities	Mandatory	Key, see 7.2.7.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.7.2   Property: ManagedElement

The presentation requirement level for this property is mandatory.

Shall reference an instance of CIM_Fan representing the fan.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation Fan.

- The multiplicity of [1 .. 1] defined in the schema is constrained to [1 .. *].**Profile Error: Reference "ManagedElement" defined in association class adaptation "ElementCapabilities" constrains its maximum multiplicity to "unbounded", which is invalid because it is greater than the maximum multiplicity "1" defined in schema association class "CIM_ElementCapabilities".**

7.2.7.3   Property: Capabilities

The presentation requirement level for this property is mandatory.

Shall reference an instance of CIM_EnabledLogicalElementCapabilities that describes the capabilities of CIM_Fan.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation FanCapabilities.

- The multiplicity of [0 .. *] defined in the schema is constrained to [0 .. 1].

7.2.8   Adaptation: CooledElement: CIM_ManagedSystemElement

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated ordinary adaptation.

A concrete subclass of the abstract schema class CIM_ManagedSystemElement needs to be implemented.

The requirement level for this adaptation is conditional, with the following condition:

> The PartialCooling feature is implemented.

Table 20 identifies the element requirements for this adaptation.

Table 20 – CooledElement: Element requirements

Element	Requirement	Description
Operations		
Associators( ) for AssociatedCooling	Mandatory	See DSP0223.
AssociatorNames( ) for AssociatedCooling	Mandatory	See DSP0223.
References( ) for AssociatedCooling	Mandatory	See DSP0223.
ReferenceNames( ) for AssociatedCooling	Mandatory	See DSP0223.

7.2.9   Adaptation: AssociatedCooling: CIM_AssociatedCooling

7.2.9.1   General

This adaptation does not define a model description.

The managed system element for which the fan provides cooling is represented by the CIM_Fan instance that is associated with a CIM_ManagedSystemElement subclass instance through the CIM_AssociatedCooling association. When no instance of CIM_AssociatedCooling references the instance of CIM_Fan, the fan represented by CIM_Fan cools the whole managed system represented by the CIM_System instance associated with the CIM_Fan instance via CIM_SystemDevice. When at least one instance of CIM_AssociatedCooling references the instance of CIM_Fan, the elements cooled by the fan shall be only those referenced by the CIM_AssociatedCooling association and not the whole managed system.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> The PartialCooling feature is implemented.

Table 21 identifies the element requirements for this adaptation.

Table 21 – AssociatedCooling: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.9.2.
Dependent	Mandatory	Key, see 7.2.9.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.9.2   Property: Antecedent

The presentation requirement level for this property is mandatory.

Shall reference an instance of CIM_Fan representing the fan.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation Fan.
- The multiplicity of [0 .. *] defined in the schema is constrained to [1 .. *].

7.2.9.3 Property: Dependent

The presentation requirement level for this property is mandatory.

Shall reference an instance of a subclass of CIM_ManagedSystemElement for which the fan is providing cooling.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation CooledElement.
- The multiplicity of [0 .. *] defined in the schema is constrained to [1 .. *].

7.2.10 Adaptation: FanRedundancySet: CIM_RedundancySet

7.2.10.1 General

This adaptation models fan redundancy groups for which the feature 'fan redundancy' is implemented.

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is conditional, with the following condition:

At least one of the following is true:

- The FanRedundancyBySparing feature is implemented.
- The FanRedundancyByBalancing feature is implemented.

Table 22 identifies the element requirements for this adaptation.

Table 22 – FanRedundancySet: Element requirements

Element	Requirement	Description
Properties		
InstanceID	Mandatory	Key, see schema definition.
RedundancyStatus	Mandatory	See schema definition.
TypeOfSet	Mandatory	See 7.2.10.2.
MinNumberNeeded	Mandatory	See 7.2.10.3.
ElementName	Mandatory	See 7.2.10.4.
Methods		
Failover( )	Optional	See 7.2.10.5.
Operations		
GetInstance( )	Mandatory	See DSP0223.
EnumerateInstances( )	Mandatory	See DSP0223.
EnumerateInstanceNames( )	Mandatory	See DSP0223.
Associators( ) for HostedRedundancySet and others	Mandatory	See 7.2.10.6.
AssociatorNames( ) for HostedRedundancySet and others	Mandatory	See 7.2.10.7.
References( ) for HostedRedundancySet and others	Mandatory	See 7.2.10.8.

Element	Requirement	Description
ReferenceNames( ) for HostedRedundancySet and others	Mandatory	See 7.2.10.9.

7.2.10.2 Property: TypeOfSet

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this property:

- OCL constraint with context of a FanRedundancySet instance:

```
inv:
mrpIsFeatureSupported('fan redundancy type \'load-balanced\'',self)
implies
    self.TypeOfSet->forall( v | Set { 3 /* Load Balanced */,
        2 /* N+1 */ }->includes(v))
```

Explanation:

> If feature 'fan redundancy type "load-balanced"' is supported for a fan redundancy group represented by a CIM_RedundancySet instance, its TypeOfSet array property shall contain the values 3 (Load Balanced), 2 (N+1), or both, and shall not contain any other values.

- OCL constraint with context of a FanRedundancySet instance:

```
inv:
mrpIsFeatureSupported('fan redundancy type \'sparing\'',self)
implies
    self.TypeOfSet->forall( v | Set { 4 /* Sparing */,
        5 /* Limited Sparing */ }->includes(v))
```

Explanation:

> If feature 'fan redundancy type "sparing"' is supported for a fan redundancy group represented by a CIM_RedundancySet instance, its TypeOfSet array property shall contain the values 4 (Sparing), 5 (Limited Sparing), or both, and shall not contain any other values.

7.2.10.3 Property: MinNumberNeeded

The presentation requirement level for this property is mandatory.

Shall match 0 when the minimum number of fans needed for the redundancy is unknown.

7.2.10.4 Property: ElementName

The presentation requirement level for this property is mandatory.

Shall be formatted as a free-form string of variable length, using the pattern ".*".

7.2.10.5 Method: Failover( )

The requirement level for this method is optional.

The Failover() method forces a failover from one member of a CIM_RedundancySet collection to another. After the successful execution of the method, the fan that is represented by the CIM_Fan instance

referenced by the FailoverFrom parameter becomes inactive. The fan that is represented by the CIM_Fan instance referenced by the FailoverTo parameter takes over as the active fan.

The Failover() method may be supported if the FailoverSupported property of at least one instance of CIM_IsSpare that references the CIM_RedundancySet instance has a value of 3 (Manual) or 4 (Both Manual and Automatic). The Failover() method shall not be supported if the FailoverSupported property of every instance of CIM_IsSpare that references the CIM_RedundancySet has a value of 2 (Automatic).

The execution of the Failover() method shall return a value of 2 (Error Occurred) under the following conditions:

- The CIM_Fan instance that is referenced by the FailoverTo parameter is not a spare fan .

- The CIM_Fan instance that is referenced by the FailoverFrom parameter is not associated with the CIM_RedundancySet only through an instance of CIM_MemberOfCollection.

After the successful execution of the Failover() method, the following actions occur:

- The CIM_Fan that is referenced by the FailoverTo parameter shall take over as the active fan.

- The CIM_Fan instance that is referenced by the FailoverTo parameter shall be associated with the CIM_RedundancySet only through an instance of CIM_MemberOfCollection.

- The CIM_Fan instance that is referenced by FailoverFrom parameter shall become a spare fan .

- When fan state management is supported, the CIM_Fan instance that is referenced by the FailoverFrom parameter shall not have an EnabledState property value of 2 (Enabled).

Table 23 identifies the parameter and return value requirements of the method.

Table 23 – Failover( ): Parameter requirements

Parameter	Description
FailoverFrom	In, see 7.2.10.5.1.
FailoverTo	In, see 7.2.10.5.2.
ReturnValue	See 7.2.10.5.3.

7.2.10.5.1 Parameter: FailoverFrom

The redundant fan that will become inactive.

The implementation shall satisfy the following constraint for this reference parameter:

      Referenced instances shall be of class adaptation Fan.

7.2.10.5.2 Parameter: FailoverTo

The redundant fan that will become active and take over for the inactivated fan.

The implementation shall satisfy the following constraint for this reference parameter:

      Referenced instances shall be of class adaptation Fan.

7.2.10.5.3 Return value

This method shall return one of the following return values:

Table 24 – Failover: Return values

Value	Description
0	The failover was successfully performed.
1	The method is not implemented.
2	An error has occurred.

7.2.10.6   Operation: Associators( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet
- IsSpare
- MemberOfRedundancySet
- OwningCollectionElement

7.2.10.7   Operation: AssociatorNames( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet
- IsSpare
- MemberOfRedundancySet
- OwningCollectionElement

7.2.10.8   Operation: References( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet
- IsSpare
- MemberOfRedundancySet
- OwningCollectionElement

7.2.10.9   Operation: ReferenceNames( ) for HostedRedundancySet and others

For general requirements on the implementation of this operation, see DSP0223.

The requirement level for this operation is mandatory.

This operation requirement applies when traversing the following association adaptations:

- HostedRedundancySet

- IsSpare
- MemberOfRedundancySet
- OwningCollectionElement

7.2.11 Adaptation: OwningCollectionElement: CIM_OwningCollectionElement

7.2.11.1 General

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> At least one of the following is true:
>
>> - The FanRedundancyBySparing feature is implemented.
>> - The FanRedundancyByBalancing feature is implemented.

Table 25 identifies the element requirements for this adaptation.

Table 25 – OwningCollectionElement: Element requirements

Element	Requirement	Description
Properties		
OwningElement	Mandatory	Key, see 7.2.11.2.
OwnedElement	Mandatory	Key, see 7.2.11.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.11.2 Property: OwningElement

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation ComputerSystem.
- The multiplicity of [0 .. 1] defined in the schema is constrained to [1 .. 1].

7.2.11.3 Property: OwnedElement

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation FanRedundancySet.
- The multiplicity of [0 .. *] defined in the schema is not further constrained.

7.2.12 Adaptation: HostedRedundancySet: CIM_HostedCollection

7.2.12.1 General

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> At least one of the following is true:
>> • The FanRedundancyBySparing feature is implemented.
>>
>> • The FanRedundancyByBalancing feature is implemented.

Table 26 identifies the element requirements for this adaptation.

Table 26 – HostedRedundancySet: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.12.2.
Dependent	Mandatory	Key, see 7.2.12.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.12.2 Property: Antecedent

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

> • Referenced instances shall be of class adaptation ComputerSystem.
>
> • The multiplicity of [1 .. 1] defined in the schema is not further constrained.

7.2.12.3 Property: Dependent

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

> • Referenced instances shall be of class adaptation FanRedundancySet.
>
> • The multiplicity of [0 .. *] defined in the schema is not further constrained.

7.2.13 Adaptation: MemberOfRedundancySet: CIM_MemberOfCollection

7.2.13.1 General

This adaptation does not define a model description.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> At least one of the following is true:
>> • The FanRedundancyBySparing feature is implemented.
>>
>> • The FanRedundancyByBalancing feature is implemented.

Table 27 identifies the element requirements for this adaptation.

Table 27 – MemberOfRedundancySet: Element requirements

Element	Requirement	Description
Properties		
Collection	Mandatory	Key, see 7.2.13.2.
Member	Mandatory	Key, see 7.2.13.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.13.2 Property: Collection

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation FanRedundancySet.
- The multiplicity of [0 .. *] defined in the schema is not further constrained.

7.2.13.3 Property: Member

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation Fan.
- The multiplicity of [0 .. *] defined in the schema is constrained to [1 .. *].

7.2.14 Adaptation: IsSpare: CIM_IsSpare

7.2.14.1 General

This adaptation models the relationship between spare fans and their redundancy group.

The implementation type of this adaptation is instantiated association adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> The FanRedundancyBySparing feature is implemented.

The implementation shall satisfy the following constraints for this adaptation:

- OCL constraint with context of a IsSpare instance:

```
inv:
self.Antecedent.EnabledState = 3 /* Disabled */
implies
        self.SpareStatus = 3 /* Cold Standby */
```

    Explanation:

> If the CIM_Fan instance (representing the spare fan) referenced from this association
> instance has an EnabledState property value of 3 (Disabled), the value of the referencing
> CIM_IsSpare instance's SpareStatus property shall be 3 (Cold Standby).

- OCL constraint with context of a IsSpare instance:

```
inv:
self.Antecedent.EnabledState != 3 /* Disabled */
implies
        self.SpareStatus = 0 /* Unknown */
```

Explanation:

> If the CIM_Fan instance (representing the spare fan) referenced from this association instance has an EnabledState property value other than 3 (Disabled), the value of the referencing CIM_IsSpare instance's SpareStatus property shall be 0 (Unknown).

Table 28 identifies the element requirements for this adaptation.

Table 28 – IsSpare: Element requirements

Element	Requirement	Description
Properties		
Antecedent	Mandatory	Key, see 7.2.14.2.
Dependent	Mandatory	Key, see 7.2.14.3.
Operations		
GetInstance( )	Mandatory	See DSP0223.

7.2.14.2 Property: Antecedent

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation Fan.
- The multiplicity of [0 .. *] defined in the schema is constrained to [1 .. *].

7.2.14.3 Property: Dependent

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraints for this reference property:

- Referenced instances shall be of class adaptation FanRedundancySet.
- The multiplicity of [0 .. *] defined in the schema is not further constrained.

7.2.15 Adaptation: NumericFanSpeedSensor: CIM_NumericSensor

7.2.15.1 General

This adaptation models analog speed sensors .

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> The FanSpeedSensor feature is implemented.

Design Note: DSP1013 1.0 defines the requirement level as optional. However, clause 7.10 of DSP1013 1.0 states the condition, so this machine readable profile has consistently defined the requirement level to be conditional.

Table 29 identifies the element requirements for this adaptation.

Table 29 – NumericFanSpeedSensor: Element requirements

Element	Requirement	Description
Base adaptations		
SpeedSensors::NumericSensor	Optional	See SpeedSensors::NumericSensor.
Properties		
SensorType	Mandatory	See 7.2.15.2.
BaseUnits	Mandatory	See 7.2.15.3.
RateUnits	Mandatory	See 7.2.15.4.
CurrentReading	Mandatory	See schema definition.

7.2.15.2 Property: SensorType

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a NumericFanSpeedSensor instance:

```
inv:
self.SensorType = 5 /* Tachometer */
```

Explanation:

The value of the SensorType property shall be 5 (Tachometer).

7.2.15.3 Property: BaseUnits

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a NumericFanSpeedSensor instance:

```
inv:
self.BaseUnits = 19 /* RPM */
```

Explanation:

The value of the BaseUnits property shall be 19 (RPM).

7.2.15.4 Property: RateUnits

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a NumericFanSpeedSensor instance:

```
inv:
self.RateUnits = 0 /* None */
```

Explanation:

The value of the RateUnits property shall be 0 (None).

7.2.16   Adaptation: DiscreteFanSpeedSensor: CIM_Sensor

7.2.16.1   General

This adaptation models discrete fan speed sensors .

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> The FanSpeedSensor feature is implemented.

Design Note: DSP1013 1.0 defines the requirement level as optional. However, clause 7.10 of DSP1013 1.0 states the condition, so this machine readable profile has consistently defined the requirement level to be conditional.

Table 30 identifies the element requirements for this adaptation.

Table 30 – DiscreteFanSpeedSensor: Element requirements

Element	Requirement	Description
Base adaptations		
SpeedSensors::DiscreteSensor	Optional	See SpeedSensors::DiscreteSensor.
Properties		
SensorType	Mandatory	See 7.2.16.2.

7.2.16.2   Property: SensorType

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

> OCL constraint with context of a DiscreteFanSpeedSensor instance:

```
inv:
self.SensorType = 5 /* Tachometer */
```

> Explanation:

> The value of the SensorType property shall be 5 (Tachometer).

7.2.17   Adaptation: FanAddedIndication: CIM_InstCreation

This adaptation does not define a model description.

Design Note: This adaptation has been defined to demonstrate the definition of lifecycle indications; it is not part of DSP1013 1.0.

The implementation type of this adaptation is indication.

The requirement level for this adaptation is conditional, with the following condition:

> The FanIndications feature is implemented.

Table 31 identifies the element requirements for this adaptation.

Table 31 – FanAddedIndication: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::LifecycleIndication	Mandatory	See Indications::LifecycleIndication.

7.2.18 Adaptation: FanRemovedIndication: CIM_InstDeletion

This adaptation does not define a model description.

Design Note: This adaptation has been defined to demonstrate the definition of lifecycle indications; it is not part of DSP1013 1.0.

The implementation type of this adaptation is indication.

The requirement level for this adaptation is conditional, with the following condition:

> The FanIndications feature is implemented.

Table 32 identifies the element requirements for this adaptation.

Table 32 – FanRemovedIndication: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::LifecycleIndication	Mandatory	See Indications::LifecycleIndication.

7.2.19 Adaptation: FanHealthIndication: CIM_AlertIndication

This adaptation models alert indications for reporting the health state of fans.

Design Note: This adaptation has been defined to demonstrate the definition of alert indications; it is not part of DSP1013 1.0.

Indications related to the redundancy of fans are defined in the FanRedundancyIndication adaptation.

The implementation type of this adaptation is indication.

The requirement level for this adaptation is conditional, with the following condition:

> The FanIndications feature is implemented.

Table 33 identifies the element requirements for this adaptation.

Table 33 – FanHealthIndication: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::AlertIndication	Mandatory	See Indications::AlertIndication.
Alert messages		
Platform::PLAT0458	Mandatory	Fan failed
Platform::PLAT0459	Mandatory	Fan return to OK
Platform::PLAT0460	Mandatory	Fan degraded
Platform::PLAT0462	Mandatory	Fan speed high
Platform::PLAT0463	Mandatory	Fan speed normal

7.2.20   Adaptation: FanRedundancyIndication: CIM_AlertIndication

This adaptation models alert indications related to the redundancy of fans.

Design Note: This adaptation has been defined to demonstrate the definition of alert indications; it is not part of DSP1013 1.0.

The implementation type of this adaptation is indication.

The requirement level for this adaptation is conditional, with the following condition:

> All of the following is true:
>
> - The FanIndications feature is implemented.
> - At least one of the following is true:
>   - The FanRedundancyByBalancing feature is implemented.
>   - The FanRedundancyBySparing feature is implemented.

Table 34 identifies the element requirements for this adaptation.

Table 34 – FanRedundancyIndication: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::AlertIndication	Mandatory	See Indications::AlertIndication.
Alert messages		
Platform::PLAT0452	Mandatory	Fan Redundacy Lost (sufficient)
Platform::PLAT0454	Mandatory	Fan Redundancy Lost (insufficient)
Platform::PLAT0455	Mandatory	Fan Redundancy Restored

7.2.21   Adaptation: FanSpeedAlertIndicationFilter: CIM_IndicationFilter

7.2.21.1   General

This adaptation does not define a model description.

A static filter for the fan speed related alert indications.

Design Note: This adaptation has been defined to demonstrate the definition of an indication specific (i.e. static) indication filter for certain alert indications; it is not part of DSP1013 1.0.

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is conditional, with the following condition:

> The FanIndications feature is implemented.

Table 35 identifies the element requirements for this adaptation.

Table 35 – FanSpeedAlertIndicationFilter: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::IndicationSpecificIndicationFilter	Mandatory	See Indications::IndicationSpecificIndicationFilter.
Properties		
Name	Mandatory	Key, see 7.2.21.2.

Element	Requirement	Description
Query	Mandatory	Required, see 7.2.21.3.
QueryLanguage	Mandatory	Required, see 7.2.21.4.

7.2.21.2 Property: Name

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a FanSpeedAlertIndicationFilter instance:

```
inv: self.Name = 'DMTF:Fan:SpeedAlertIndicationFilter'
```

7.2.21.3 Property: Query

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a FanSpeedAlertIndicationFilter instance:

```
inv: self.Query = 'SELECT * FROM CIM_AlertIndication WHERE MessageID IN (
"PLAT0462", "PLAT0463")'
```

7.2.21.4 Property: QueryLanguage

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

OCL constraint with context of a FanSpeedAlertIndicationFilter instance:

```
inv: self.QueryLanguage = 'DMTF:CQL'
```

7.2.22 Adaptation: FanAddedLifecycleIndicationFilter: CIM_IndicationFilter

7.2.22.1 General

This adaptation does not define a model description.

A static filter for the fan added lifecycle indication.

Design Note: This adaptation has been defined to demonstrate the definition of an indication specific (i.e. static) indication filter for a single lifecycle indication; it is not part of DSP1013 1.0.

The implementation type of this adaptation is instantiated ordinary adaptation.

The requirement level for this adaptation is conditional, with the following condition:

The FanIndications feature is implemented.

Table 36 identifies the element requirements for this adaptation.

Table 36 – FanAddedLifecycleIndicationFilter: Element requirements

Element	Requirement	Description
Base adaptations		
Indications::IndicationSpecificIndicationFilter	Mandatory	See Indications::IndicationSpecificIndicationFilter.
Properties		
Name	Mandatory	Key, see 7.2.22.2.
Query	Mandatory	Required, see 7.2.22.3.
QueryLanguage	Mandatory	Required, see 7.2.22.4.

7.2.22.2 Property: Name

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

> OCL constraint with context of a FanAddedLifecycleIndicationFilter instance:

```
inv: self.Name = 'DMTF:Fan:AddedLifecycleIndicationFilter'
```

7.2.22.3 Property: Query

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

> OCL constraint with context of a FanAddedLifecycleIndicationFilter instance:

```
inv: self.Query = 'SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA
CIM_Fan'
```

7.2.22.4 Property: QueryLanguage

The presentation requirement level for this property is mandatory.

The implementation shall satisfy the following constraint for this property:

> OCL constraint with context of a FanAddedLifecycleIndicationFilter instance:

```
inv: self.QueryLanguage = 'DMTF:CQL'
```

8   Use cases and state descriptions

8.1   State description: ObjectDiagram

Section 9.1 of Fan Profile (DSP1013) would be inserted here.

8.2   Use case: SetFanSpeed

Section 9.2 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1. Steps from Section 9.2

8.3   Use case: ResetFan

Section 9.3 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1.   Steps from Section 9.3

8.4   Use case: GetFanRedundancyStatus

Section 9.4 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1.   Steps from Section 9.4

8.5   Use case: FindSpareFan

Section 9.5 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1.   Steps from Section 9.5

8.6   Use case: ShowFanSensorInfo

Section 9.6 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1.   Steps from Section 9.6

8.7   Use case: FindCooledElements

Section 9.7 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1.   Steps from Section 9.7

8.8   Use case: DetermineElementNameModifiability

Section 9.8 of Fan Profile (DSP1013) would be inserted here.

The main flow for this use case consists of the following step:

1.   Steps from Section 9.8

ANNEX A
(informative)

Change log

Table 37 – Change log

Version	Date	Description
1.0.0a	2006-06-13	DSP1013: Released as a Preliminary Standard
1.0.0	2007-10-12	DSP1013: Released as a Final Standard
1.0.1	2008-09-23	DSP1013: Released as a Final Standard
1.0.2m	2011-08-31	XMP1013: Included as a sample profile into DSP2023

Bibliography

This clause lists references that are helpful for the application of this document.

DMTF DSP1000, *Management Profile Specification Template 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1000_1.1.pdf