Document Number: DSP1053

Date: 2009-06-16

Version: 1.0.0

Base Metrics Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

CONTENTS

Figures

Tables

Foreword

The *Base Metrics Profile* (DSP1053) was prepared by the Applications Working Group of the DMTF.

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

Acknowledgments

The authors wish to acknowledge the following people.

Editors:

- Oliver Benke – IBM
- Aaron Merkin – IBM

Contributors:

- Karl Schopmeyer – The Open Group
- Andreas Maier –- IBM
- Khachatur Papanyan – Dell

Introduction

The information in this specification should be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to represent and manage the components described in this document.

The target audience for this specification is implementers who are writing Common Information Model (CIM)-based providers or consumers of management interfaces that need to dynamically add metrics to existing components.

Base Metrics Profile

1 Scope

The *Base Metrics Profile* is a component profile that defines the minimum object model needed to provide dynamic metrics associated to existing managed elements and related associations. This profile does not document how to model metrics for capacity planning or accounting purposes. These topics are covered by the *Capacity Metrics Profile* (DSP1073), which is a specialization of this 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 DSP0004, *CIM Infrastructure Specification 2.3*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

DMTF DSP0200, *CIM Operations over HTTP 1.2*,
http://www.dmtf.org/standards/published_documents/DSP0200_1.2.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

DMTF DSP1033, *Profile Registration Profile 1.0*,
http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

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

referencing profile

indicates a profile that owns the definition of this class and can include a reference to this profile in its “Referenced Profiles” table

3.9

shall

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

3.10

shall not

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

3.11

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

3.12

should not

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

3.13

unspecified

indicates that this profile does not define any constraints for the referenced CIM element or operation

3.14

aggregation metric

a type of metric that is derived by applying a formula or filter to a set of base metric values

3.15

base metric

a metric provided directly without a dependency on other metric values

3.16

measured resource

a managed object being measured, which is the resource to which base metric value instances are associated

3.17

sampling interval

a value that determines how often new metric values are retrieved, if metrics are retrieved periodically

3.18

current data

the most current data available for a given metric. Online monitoring (3.19) and snapshot monitoring (3.20) are types of current data access.

3.19

online monitoring

the process in which metric values (typically interval metrics) are gathered asynchronously to a request from the instrumentation or reporting layer

3.20

snapshot monitoring

the process in which metric values are gathered synchronously with a request from the instrumentation or reporting layer

3.21

long-term monitoring

the process in which metric values are captured during an interval

3.22

event-based monitoring

the process in which threshold values for metrics are used to trigger asynchronous notification

3.23

instantaneous metrics

metrics that apply to a particular point in time. An example of an instantaneous metric is the amount of memory currently allocated to a virtual server.

3.24

interval metrics

metrics that apply to a time interval. An example of an interval metric is the average CPU utilization of a server over the past hour.

3.25

summation metrics

a type of counter metric that reflects the accumulation of a value

3.26

watermark metrics

a type of aggregation metric used to capture the minimum or maximum value recorded for a monitored value

4 Symbols and Abbreviated Terms

4.1
CPU
central processing unit

4.2
IEPS
instructions executed per second

4.3
UTC
Universal Time Coordinated

4.4
UUID
Universally Unique Identifier

5 Synopsis

Profile Name: Base Metrics

Version: 1.0.0

Organization: DMTF

CIM Schema Version: 2.22

Central Class: CIM_MetricService

Scoping Class: CIM_System

Table 1 identifies profiles on which this profile has a dependency.

Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

6 Description (Informative)

The Metrics Model provides the ability to model and control metrics captured for managed elements.

Figure 1 represents the class schema for the *Base Metrics Profile*. For simplicity, the prefix CIM_ has been removed from the names of the classes.

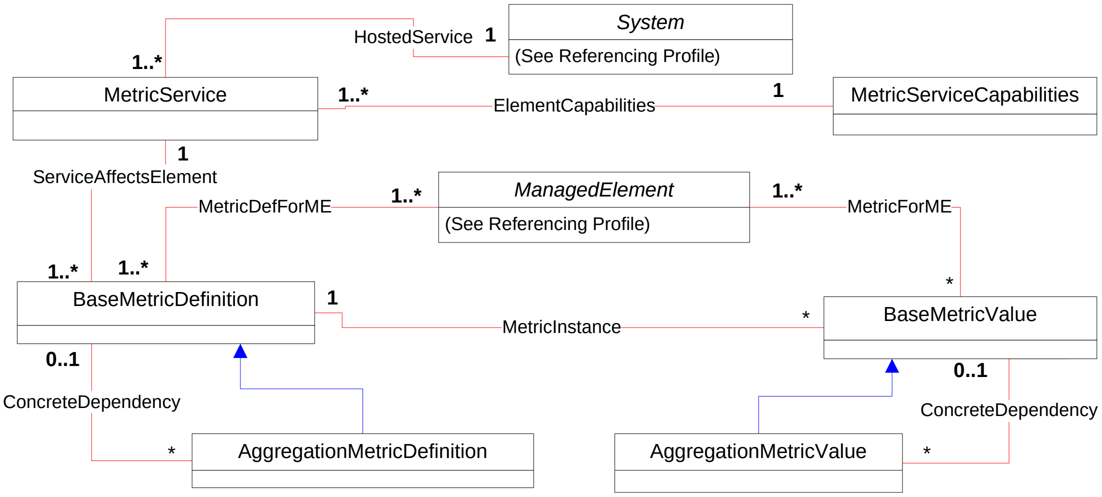


Figure 1 – Base Metrics Profile: Class Diagram

A metric instance is represented by an instance of CIM_BaseMetricValue or its subclass CIM_AggregationMetricValue. The definition of the metric is provided by an associated instance of CIM_BaseMetricDefinition or CIM_AggregationMetricDefinition. The context of the metric is provided by one or more associated instances of CIM_ManagedElement. For example, an instance of CIM_ManagedElement could represent an operating system, a cluster, or a complex software application containing application server and database server parts. The modeling of the associated resources is out of the scope of this profile.

When defining a metric, there are four main characteristics to consider:

- Metric access type
- Time scope of the metric
- Formulation of the metric value
- Metric context

These characteristics are described in the following sections. Some of these characteristics are modeled as attributes of an instance of CIM_BaseMetricDefinition. Others are modeled through the relationship of an instance of CIM_BaseMetricDefinition or CIM_BaseMetricValue to one or more instances of CIM_ManagedElement.

6.1 Metric Access Types

There are three major access types for metrics and performance data:

- Current data access, for data gathered in the recent past
- Long-term monitoring, for historical time series data
- Event-based monitoring, for asynchronous indication subscriptions based on instances of CIM_BaseMetricValue

6.1.1 Current Data

Current data access is the most common access type for dynamic metrics. The purpose is to request the most current data available to the implementation. There are two paradigms for the gathering metrics with an access type of current data, online monitoring and snapshot monitoring.

For the current data access type, CIM_BaseMetricValue.Volatile is equal to TRUE. The metric value property is updated at the point in time that the instance is read.

6.1.1.1 Online Monitoring

For the online monitoring access type, the CIM metric values are updated independently by the gathering infrastructure. When a new metric value is requested, the most current value is presented. Typically, the implementation of the gathering and reporting components can be separated. It is recommended to synchronize metric retrieval in order to allow for correlation of various metrics. For the online monitoring access type, the value of the CIM_BaseMetricDefinition.GatheringType property is 3 (Periodic) or 2 (OnChange).

A well known UNIX application that implements this access type is “top”.

6.1.1.2 Snapshot Monitoring

For the snapshot monitoring access type, the CIM metric value is determined each time a client application requests a new metric value. The value of the CIM_BaseMetricDefinition.GatheringType property is 4 (OnRequest).

Note that this access type has disadvantages. For example, data generated by snapshot monitoring is not always suitable for event correlation. However, for simple investigations of the current state of the system, snapshot monitoring is suitable, and it has the advantage that the gathering infrastructure needs to be active only on request rather than continuously.

A well known UNIX application that implements this access type is "ps".

6.1.2 Long-Term Monitoring

The long-term monitoring access type is used for historical time series. For example, it could be used to collect all metric values gathered between 9:00 A.M. and 5:00 P.M. with 15 minute intervals.

For the long-term monitoring access type, the value of the CIM_BaseMetricValue.Volatile property is FALSE. The metric value is stored in a repository and can be retrieved by client applications later on.

A well known UNIX application that implements this access type is "sar/sadc".

Important aspects of the long-term monitoring access type are described in the *Capacity Metrics Profile* (DSP1073).

6.1.3 Event-Based Monitoring

The event-based monitoring access type is used for asynchronous indication subscriptions based on base metric value instances, which allows the client to subscribe for certain threshold conditions. This may be implemented based on CIM_InstModification subscriptions for CIM_BaseMetricValue changes.

Details on how to use event-based monitoring with dynamic metrics are out of the scope of this document.

6.2 Metric Time Scope

Many common types of metrics can be captured. Metrics may be quantified along two axes. The first axis is the time scope, and the second axis is the type of value formulation. Along the time scope axis, metrics can be described as instantaneous, interval, or startup interval. Types of values captured include minimum, maximum, average, instantaneous, and aggregate values.

6.2.1 Instantaneous Metrics

Instantaneous metrics report a monitored value at a given instant. An example of an instantaneous metric is the amount of power being consumed by a system at a given point in time. For instantaneous metrics, the value of the CIM_BaseMetricDefinition.TimeScope property is 2 (Point).

6.2.2 Interval Metrics

Interval metrics are metrics captured over an interval in time. Interval metrics can report values such as the average utilization of a resource over a period of time. An example of an iInterval metric is the average power consumption of a server over the last three days. For interval metrics, the value of the CIM_BaseMetricDefinition.TimeScope property is 3 (Interval).

6.2.3 Startup Interval Metrics

Startup interval metrics are metrics captured over an interval in time, for which the start of the interval is tied to a lifecycle change (initialization or creation) of the managed element for which the value is captured. An example of a startup interval metric is the total number of CPU cycles consumed for a transaction that is recorded from the time the transaction begins.

6.3 Metric Value Formulation

A metric’s value may be constructed in innumerable ways. Three common types of metrics are simple metrics, summation metrics, and aggregation metrics. These types are described in more detail in the following clauses.

6.3.1 Simple Metrics

Simple metrics report status recorded at some point in time without requiring a calculation or function to be applied to produce the value. An example of a simple metric is an instantaneous reading of the power being consumed by a server.

6.3.2 Summation Metrics

Summation metrics are used to report aggregate or total values for a monitored entity. Uses of summation metrics include billing, accounting, and capacity planning. An example of a summation metric is the total power consumed by a server for the last three days.

More information on using summation metrics is specified in DSP1073.

6.3.3 Aggregation Metrics

Aggregation metrics are metrics derived by applying a formula or filter to a set of base metric values. Aggregation metrics that apply a formula to metric values of multiple types are out of scope of this profile. The definition of an aggregation metric is provided by an instance of CIM_AggregationMetricDefinition. An aggregation metric includes the definition of a base metric as well as the function used to create the derived value. A server-side implementation may support the collection of an aggregation metric without supporting the collection of the base metric. If collection of the base metric is supported, a distinct instance of CIM_BaseMetricDefinition is used to define the base metric and distinct instances of CIM_BaseMetricValue are used to represent the metric value. The CIM_BaseMetricDefinition instance may be associated to the CIM_AggregationMetricDefinition instance, and the CIM_BaseMetricValue instance may be associated with the CIM_AggregationMetricValue instance.

6.3.3.1 Watermark Metrics

Watermark metrics are a class of aggregation metrics. A watermark metric captures the highest or lowest value recorded for a monitored entity. An example of a high watermark metric is the peak instantaneous power consumed by a server in the past hour.

6.4 Metric Context

Generally it is necessary to understand the context of a metric in order to properly interpret and utilize the reported values. An example is a metric that reports the number of packet errors per minute. If the metric is reported for a single network interface, a much lower value is a cause for concern than if the metric is for an entire network segment.

The CIM_MetricForME and CIM_MetricDefForME associations are used to provide the context in which a metric is captured. CIM_MetricDefForME associates an instance of CIM_BaseMetricDefinition with an instance of CIM_ManagedElement. This indicates that the metric defined by the CIM_BaseMetricDefinition can be captured for the resource modeled with the instance of CIM_ManagedElement. The same metric can be available for multiple instances of CIM_ManagedElement simultaneously. Therefore, it is necessary to further disambiguate the specific instance of CIM_ManagedElement for which a particular instance of the metric has been captured. The CIM_MetricForME association is used to associate an instance of CIM_BaseMetricValue with the instances of CIM_ManagedElement that provide its context.

A given defined metric may have multiple values available concurrently for a CIM_ManagedElement instance. The BreakdownValue and BreakdownDimension properties are used to differentiate among the instances of CIM_BaseMetricValue that provide multiple concurrent metric values for a CIM_ManagedElement. An example of when multiple metric values for the same metric definition may be available is when a total value and values per component exist.

7 Implementation

This section details the requirements related to the arrangement of instances and their properties for implementations of this profile.

7.1 Common Requirements

This section details the common requirements for modeling metrics. The requirements stated in this section for the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall also apply to the CIM_AggregationMetricDefinition and CIM_AggregationMetricValue subclasses, respectively.

7.1.1 Service and Capabilities

At least one instance of CIM_MetricService shall exist. Each instance of CIM_MetricService shall be associated with exactly one instance of CIM_System through the CIM_HostedService association. Each instance of CIM_MetricService shall be associated with exactly one instance of CIM_MetricServiceCapabilities through the CIM_ElementCapabilities association. Each instance of CIM_BaseMetricDefinition shall be associated with exactly one instance of CIM_MetricService through the CIM_ServiceAffectsElement association.

7.1.2 Relating a Metric Definition and Metric Value

Each instance of CIM_BaseMetricValue shall be associated with exactly one instance of CIM_BaseMetricDefinition through the CIM_MetricInstance association.

Each instance of CIM_AggregationMetricValue shall be associated with exactly one instance of CIM_AggregationMetricDefinition through the CIM_MetricInstance association.

7.1.3 Identifying a Metric Definition

Incorporating profiles will specify metric definitions for metrics that are applicable to the management domain of the incorporating profile. Each metric definition shall be uniquely identified by the Name property of the CIM_BaseMetricDefinition instance that correlates to the metric definition.

If the incorporating profile is a DMTF Management Profile, the CIM_BaseMetricDefinition.Name shall be formatted as follows:

“CIM:DSP” <DSP#> ":" <unique identifier>

<DSP#> is the document number that the DMTF has assigned to the profile, and <unique identifier> is a string value unique within the scope of the defining profile.

If the incorporating profile is not a DMTF Management Profile, the CIM_BaseMetricDefinition.Name property shall be formatted as follows:

> < OrgID > : < LocalID >, where < OrgID > and < LocalID > are separated by a colon (:) and < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the business entity that is creating or defining the value or that is a registered ID assigned to the business entity by a recognized global authority. In addition, to ensure uniqueness, < OrgID > shall not contain a colon (:). If this algorithm is used, the first colon to appear in the value shall appear between < OrgID > and < LocalID >. < LocalID > is chosen by the business entity and shall be used uniquely.

7.1.4 Identifying Metric Context

The considerations for identifying the context of a metric are provided in the following sections.

7.1.4.1 General Requirements

Each instance of CIM_BaseMetricDefinition shall be associated with at least one instance of CIM_ManagedElement through the CIM_MetricDefForME association. If the CIM_BaseMetricValue instance models a metric with the current data access type, the CIM_BaseMetricValue instance shall be associated with exactly one instance of CIM_ManagedElement through the CIM_MetricForME association.

7.1.4.2 Breakdown Dimensions (Optional)

If multiple instances of CIM_BaseMetricValue are available concurrently for a given instance of CIM_ManagedElement, where the instances of CIM_BaseMetricValue are associated with the same instance of CIM_BaseMetricDefinition through instances of the CIM_MetricValue association and the time frame for which the metric values are recorded overlaps in whole or in part, the requirements specified in this subclause shall be met.

The CIM_BaseMetricDefinition.BreakdownDimensions property shall not be NULL.

At most, one instance of CIM_BaseMetricValue may have null values for the BreakdownDimension and BreakdownValue properties.

If the incorporating profile that specifies the CIM_BaseMetricDefinition is a DMTF Management Profile, and a value of the CIM_BaseMetricDefinition.BreakdownDimensions identifies a CIM class, the value shall be formatted as:

<schemaName>"_"<simpleClassName>

as specified in DSP0004.

If the incorporating profile that specifies the CIM_BaseMetricDefinition is a DMTF Management Profile, and a value of the CIM_BaseMetricDefinition.BreakdownDimensions does not identify a CIM class, the value shall be formatted as follows:

“CIM:DSP” <DSP#> ":" <unique identifier>

<DSP#> is the document number that the DMTF has assigned to the profile, and <unique identifier> is a string value unique within the scope of the defining profile.

If the incorporating profile is not a DMTF Management Profile, each value of the CIM_BaseMetricDefinition.BreakdownDimensions property shall be formatted as follows:

> < OrgID > : < LocalID >, where < OrgID > and < LocalID > are separated by a colon (:) and < OrgID > shall include a copyrighted, trademarked, or otherwise unique name that is owned by the business entity that is creating or defining the value or that is a registered ID assigned to the business entity by a recognized global authority. In addition, to ensure uniqueness, < OrgID > shall not contain a colon (:). If using this algorithm, the first colon to appear in the value shall appear between < OrgID > and < LocalID >. < LocalID > is chosen by the business entity and shall be used uniquely.

If the CIM_BaseMetricValue.BreakdownValue identifies a CIM instance, the CIM_BaseMetricValue.BreakdownValue property shall be formatted as a WBEM URI (as defined in DSP0207) that identifies the CIM instance.

If the value of the CIM_BaseMetricValue.BreakdownDimension property is not NULL, it shall be one of the values contained in the CIM_BaseMetricDefinition.BreakdownDimensions property of the associated instance of CIM_BaseMetricDefinition. If the CIM_BaseMetricValue.BreakdownDimension property is NULL, the CIM_BaseMetricValue.BreakdownValue property shall be NULL.

7.1.5 Gathering Type

If values for an instance of CIM_BaseMetricDefinition are gathered through online monitoring, the CIM_BaseMetricDefinition.GatheringType property shall have a value of 3 (Periodic) or 2 (OnChange). If values for an instance of CIM_BaseMetricDefinition are gathered through snapshot monitoring, the CIM_BaseMetricDefinition.GatheringType property shall have a value of 4 (OnRequest).

7.2 Modeling Metric Access Types

This section details requirements for modeling different metric access types. The requirements stated in this section for the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall also apply to the CIM_AggregationMetricDefinition and CIM_AggregationMetricValue subclasses, respectively.

7.2.1 Modeling Current Data Access Type (Optional)

Metrics with an access type of current data may be supported. If metrics with an access type of current data are modeled, the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall be used as defined in 10.10 and 10.12, respectively.

7.3 Modeling Metric Time Scope

This section details requirements for modeling metrics with common time scopes. The requirements stated in this section for CIM_BaseMetricDefinition and CIM_BaseMetricValue shall also apply to the CIM_AggregationMetricDefinition and CIM_AggregationMetricValue subclasses, respectively.

7.3.1 Modeling Instantaneous Metrics (Optional)

Instantaneous metrics may be modeled. If instantaneous metrics are modeled, the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall be used as defined in 10.6 and 10.11, respectively.

7.3.2 Modeling Interval Metrics (Optional)

Interval metrics may be modeled. If interval metrics are modeled, the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall be used as defined in 10.7 and 10.13, respectively.

7.3.3 Modeling Interval Metrics (Optional)

Startup interval metrics may be modeled. If interval metrics are modeled, the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall be used as defined in 10.8 and 10.14, respectively.

7.4 Modeling Metric Value Formulation

This section details requirements for modeling metrics with common value formulations.

7.4.1 Modeling Summation Metrics (Optional)

Summation metrics may be modeled. If summation metrics are modeled, the CIM_BaseMetricDefinition and CIM_BaseMetricValue clases shall be used as defined in 10.9 and 10.15, respectively.

7.4.2 Modeling Aggregation Metrics (Optional)

Aggregation metrics may be modeled. When aggregation metrics are modeled, the requirements specified in this section shall be met. An instance of CIM_AggregationMetricDefinition shall define the aggregation metric. An instance of CIM_AggregationMetricValue shall exist for each aggregation metric value.

7.4.2.1 Modeling Low Watermark Metrics (Optional)

If a low watermark metric is modeled, the instance of CIM_AggregationMetricDefinition that defines the metric shall be implemented as defined in 10.2.

7.4.2.2 Modeling High Watermark Metrics (Optional)

If a high watermark metric is modeled, the instance of CIM_AggregationMetricDefinition that defines the metric shall be implemented as defined in 10.3.

7.5 Relationship between Aggregation and Base Metrics

If an aggregation metric that is defined by an instance of CIM_AggregationMetricDefinition reports a value derived from a base metric that is modeled with an instance of CIM_BaseMetricDefinition, the instance of CIM_AggregationMetricDefinition may be associated with the instance of CIM_BaseMetricDefinition through an instance of CIM_ConcreteDependency, where the instance of CIM_ConcreteDependency is as defined in 10.17. If the aggregation metric value modeled with an instance of CIM_AggregationMetricValue is identical to a base metric value for the base metric definition from which the aggregation metric is derived, the instance of CIM_AggregationMetricValue may be associated with the CIM_BaseMetricValue through an instance of CIM_ConcreteDependency that is implemented as defined in 10.18.

7.6 Constraints on Metric Values for Controllable Metrics

The ability to control the collection of a metric defined by an instance of CIM_BaseMetricDefinition for a managed element represented by an instance of CIM_ManagedElement may be supported.

If the value of the MetricCollectionEnabled property of the CIM_MetricDefForME instance that associates an instance of CIM_BaseMetricDefinition with an instance of CIM_ManagedElement has the value 3 (Disabled), an instance of CIM_BaseMetricValue shall not be associated with the CIM_BaseMetricDefinition through CIM_MetricInstance where the instance of CIM_BaseMetricValue is associated with the CIM_ManagedElement instance through CIM_MetricForME and the value of the CIM_BaseMetricValue.Volatile property is 2 (Enabled).

The value of the RecordedSince property of an instance of CIM_MetricDefForME shall not reflect a value earlier in time than the time when the MetricCollectionEnabled property of the instance of CIM_MetricDefForME last transitioned from a value of 3 (Disabled) to 2 (Enabled).

For an instance of CIM_BaseMetricValue that is associated with an instance of CIM_BaseMetricDefinition through CIM_MetricInstance and that is associated with an instance of CIM_ManagedElement through the CIM_MetricForME association, if an instance of CIM_BaseMetricValue has a value of 2 (Enabled) for the Volatile property, the value of the TimeStamp property or the value calculated by subtracting the value of the Duration property from the value of the TimeStamp property shall not specify a point in time earlier than the value of the RecordedSince property of the instance of CIM_MetricDefForME that associates the instance of CIM_BaseMetricDefinition to the instance of CIM_ManagedElement.

8 Methods

This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM elements defined by this profile. For the extrinsic methods defined in clauses 8.1 through 8.5, the requirements pertaining to the CIM_BaseMetricDefinition and CIM_BaseMetricValue classes shall also apply to the CIM_AggregationMetricDefinition and CIM_AggregationMetricValue subclasses, respectively.

8.1 CIM_MetricService.ShowMetrics( )

The ShowMetrics( ) method provides the ability to query for metrics that a server-side implementation is able to collect, as well as whether or not collection of the metric is currently enabled.

The ShowMetrics( ) method’s return code values shall be as specified in Table 2 where the method execution behavior matches the return code description. The ShowMetrics( ) method’s parameters are specified in Table 3.

No standard messages are defined for this method.

Table 2 – CIM_MetricService.ShowMetrics( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 3 – CIM_MetricService.ShowMetrics( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Subject	CIM_ManagedElement REF	Reference to the CIM_ManagedElement for which metrics will be reported
IN	Definition	CIM_BaseMetricDefinition REF	Reference to the CIM_BaseMetricDefinition to query for values of
OUT	ManagedElements	CIM_ManagedElement REF[ ]	Array of references to instances of CIM_ManagedElement for which the metric identified by the Definition parameter is being collected
OUT	DefinitionList	REF[ ]	Array of references to instances of CIM_BaseMetricDefinition defining metrics being collected for the CIM_ManagedElement instance identified by the Subject parameter
OUT	MetricNames	string[ ]	Array of metric names for the instances of CIM_BaseMetricDefinition specified by the DefinitionList parameter
OUT	MetricCollectionEnabled	uint16[ ]	Array of values indicating whether or not a metric is being collected

8.1.1 CIM_MetricService.ShowMetrics( ) Conditional Support

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities contains the value 4 (ShowMetrics), the ShowMetrics( ) method shall be implemented and shall not return the value 1 (Not Supported).

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities does not contain the value 4 (ShowMetrics), the ShowMetrics( ) method shall not be implemented or shall always return the value 1 (Not Supported).

8.2 CIM_MetricService.ShowMetricsByClass( )

The ShowMetricsByClass( ) method provides the ability to query for metrics that a server-side implementation is able to collect, as well as whether or not collection of the metric is currently enabled.

The ShowMetricsByClass( ) method's return code values shall be as specified in Table 4 where the method execution behavior matches the return code description. The ShowMetricsByClass( ) method's parameters are specified in Table 5.

No standard messages are defined for this method.

Table 4 – CIM_MetricService.ShowMetricsByClass( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 5 – CIM_MetricService.ShowMetricsByClass( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Subject	CIM_ManagedElement REF	Identifies a CIM class for which metrics will be reported
IN	Definition	CIM_BaseMetricDefinition REF	Reference to the CIM_BaseMetricDefinition to query for values of
OUT	DefinitionList	REF[ ]	Array of references to instances of CIM_BaseMetricDefinition defining metrics being collected for the CIM class identified by the Subject parameter
OUT	MetricNames	string[ ]	Array of metric names for the instances of CIM_BaseMetricDefinition specified by the DefinitionList parameter
OUT	MetricCollectionEnabled	uint16[ ]	Array of values indicating whether or not a metric is being collected

8.2.1 CIM_MetricService.ShowMetricsByClass( ) Conditional Support

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities contains the value 5 (ShowMetricsByClass), the ShowMetricsByClass( ) method shall be implemented and shall not return the value 1 (Not Supported).

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities does not contain the value 5 (ShowMetricsByClass), the ShowMetricsByClass( ) method shall not be implemented or shall always return the value 1 (Not Supported).

8.3 CIM_MetricService.ControlMetrics( )

The ControlMetrics( ) method provides the ability to enable or disable the collection of:

- a metric for all instances of CIM_ManagedElement
- all metrics for a single CIM_ManagedElement instance
- a single metric for a single CIM_ManagedElement instance

The ControlMetrics( ) method’s return code values shall be as specified in Table 6 where the method execution behavior matches the return code description. The ControlMetrics( ) method’s parameters are specified in Table 7.

No standard messages are defined for this method.

Table 6 – CIM_MetricService.ControlMetrics( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 7 – CIM_MetricService.ControlMetrics( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Subject	CIM_ManagedElement REF	Reference to the CIM_ManagedElement for which metrics will be controlled
IN	Definition	CIM_BaseMetricDefinition REF	Reference to the CIM_BaseMetricDefinition for which collection is to be enabled or disabled
IN, REQ	MetricCollectionEnabled	uint16	Value indicating whether or not the metric is collected

8.3.1 CIM_MetricService.ControlMetrics( ) Conditional Support

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities contains the value 2 (ControlMetrics), the ControlMetrics( ) method shall be implemented and shall not return the value 1 (Not Supported).

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities does not contain the value 2 (ControlMetrics), the ControlMetrics( ) method shall not be implemented or shall always return the value 1 (Not Supported).

8.3.2 Parameter Validation

If the Subject parameter is NULL and the instance of CIM_BaseMetricDefinition that is identified by the Definition parameter is not identified by a value of the ControllableMetrics property of the associated instance of CIM_MetricServiceCapabilities where the corresponding array index of the MetricsControlTypes property of the CIM_MetricServiceCapabilities instance has the value 3 (Bulk) or 4 (Both), the method shall return a value of 2 (Failed).

If the Definition parameter is NULL and the instance of CIM_ManagedElement identified by the Subject parameter is not identified by a value of the ControllableManagedElement property of the associated instance of CIM_MetricServiceCapabilities where the corresponding array index of the ManagedElementControlTypes property of the CIM_MetricServiceCapabilities instance has the value 3 (Bulk) or 4 (Both), the method shall return a value of 2 (Failed).

If both the Subject and Definition parameters are non-null, the method shall return a value of 2 (Failed) if neither of the following conditions is met:

- The instance of CIM_ManagedElement identified by the Subject parameter is identified by a value of the ControllableManagedElements property of the associated instance of CIM_MetricServiceCapabilities, where the corresponding array index of the ManagedElementControlTypes property of the CIM_MetricServiceCapabilities instance has a value of 2 (Discrete), and the instance of CIM_BaseMetricDefinition identified by the Definition parameter is identified by a value of the ControllableMetrics property of the associated instance of CIM_MetricServiceCapabilities, where the corresponding array index of the MetricsControlTypes property of the CIM_MetricServiceCapabilities instance has a value of 2 (Discrete).
- The instance of CIM_BaseMetricDefinition identified by the Definition parameter is identified by a value of the ControllableMetrics property of the associated instance of CIM_MetricServiceCapabilities, where the corresponding array index of the MetricsControlTypes property of the CIM_MetricServiceCapabilities instance has a value of 2 (Discrete) and no instances of CIM_ManagedElement that are associated with the CIM_BaseMetricDefinition through the CIM_MetricDefForME are identified by a value of the ControllableManagedElements property of the associated instance of CIM_MetricServiceCapabilities.

NOTE: The effect of the second condition is to allow the advertisement of support for controlling the collection of every metric value for a CIM_BaseMetricDefinition instance without having to explicitly list each CIM_ManagedElement instance in the ControllableManagedElements property.

8.4 CIM_MetricService.ControlMetricsByClass( )

The ControlMetricsByClass( ) method provides the ability to enable or disable the collection of:

- a metric for all instances of a specific CIM class
- all metrics for all instances of a specific CIM class
- a single metric for a single CIM_ManagedElement

The ControlMetricsByClass( ) method's return code values shall be as specified in Table 8 where the method execution behavior matches the return code description. The ControlMetricsByClass( ) method's parameters are specified in Table 9.

No standard messages are defined for this method.

Table 8 – CIM_MetricService.ControlMetricsByClass( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 9 – CIM_MetricService.ControlMetricsByClass( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Subject	CIM_ManagedElement REF	Reference to the CIM class for which metrics will be controlled
IN	Definition	CIM_BaseMetricDefinition REF	Reference to the CIM_BaseMetricDefinition for which collection is to be enabled or disabled
IN, REQ	MetricCollectionEnabled	uint16	Value indicating whether the metric is to be enabled or disabled

8.4.1 CIM_MetricService.ControlMetricsByClass( ) Conditional Support

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities contains the value 3 (ControlMetricsByClass), the ControlMetricsByClass( ) method shall be implemented and shall not return the value 1 (Not Supported).

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities does not contain the value 3 (ControlMetricsByClass), the ControlMetricsByClass( ) method shall not be implemented or shall always return the value 1 (Not Supported).

8.4.2 Parameter Validation

If the Subject parameter is NULL and the instance of CIM_BaseMetricDefinition that is identified by the Definition parameter is not identified by a value of the ControllableMetrics property of the associated instance of CIM_MetricServiceCapabilities where the corresponding array index of the MetricsControlTypes property of the CIM_MetricServiceCapabilities instance has the value 3 (Bulk) or 4 (Both), the method shall return a value of 2 (Failed).

8.5 CIM_MetricService.GetMetricValues( )

The GetMetricValues( ) method provides the ability to query for metric values.

The GetMetricValues( ) method's return code values shall be as specified in Table 10 where the method execution behavior matches the return code description. The GetMetricValues( ) method's parameters are specified in Table 11.

No standard messages are defined for this method.

Table 10 – CIM_MetricService.GetMetricValues( ) Method: Return Code Values

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

Table 11 – CIM_MetricService.GetMetricValues( ) Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Definition	CIM_BaseMetricDefinition REF	Reference to the CIM_BaseMetricDefinition to query for values
IN	Range	uint16	Identifies how the values are selected
IN	Count	uint16	Identifies the maximum number of instances to return
OUT	Values	CIM_BaseMetricValue REF[ ]	Array of references to instances of CIM_BaseMetricValue corresponding to the CIM_BaseMetricValue instances that match the query constraints identified by the input parameters

8.5.1 CIM_MetricService.GetMetricValues( ) Conditional Support

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities contains the value 6 (GetMetricValues), the GetMetricValues( ) method shall be implemented and shall not return the value 1 (Not Supported).

If the SupportedMethods property array of the associated instance of CIM_MetricServiceCapabilities does not contain the value 6 (GetMetricValues), the GetMetricValues( ) method shall not be implemented or shall always return the value 1 (Not Supported).

8.6 Profile Conventions for Operations

For each profile class (including associations), the implementation requirements for operations, including those in the following default list, are specified in class-specific subclauses of this clause.

The default list of operations is as follows:

- GetInstance
- Associators
- AssociatorNames
- References

- ReferenceNames
- EnumerateInstances
- EnumerateInstanceNames

8.7 CIM_AggregationMetricDefinition

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.8 CIM_AggregationMetricValue

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.9 CIM_BaseMetricDefinition

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.10 CIM_BaseMetricValue

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

8.11 CIM_ConcreteDependency

Table 12 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 12, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 12 – Operations: CIM_ConcreteDependency

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.12 CIM_ElementCapabilities

Table 13 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 13, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 13 – Operations: CIM_ElementCapabilities

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.13 CIM_HostedService

Table 14 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 14, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 14 – Operations: CIM_HostedService

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.14 CIM_MetricDefForME

Table 15 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 15, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 15 – Operations: CIM_MetricDefForME

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.15 CIM_MetricForME

Table 16 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 16, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 16 – Operations: CIM_MetricForME

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.16 CIM_MetricInstance

Table 17 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 17, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 17 – Operations: CIM_MetricInstance

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

8.17 CIM_MetricService

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

8.18 CIM_MetricServiceCapabilities

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

8.19 CIM_ServiceAffectsElement

Table 18 lists implementation requirements for operations. If implemented, these operations shall be implemented as defined in DSP0200. In addition, and unless otherwise stated in Table 18, all operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE: Related profiles may define additional requirements on operations for the profile class.

Table 18 – Operations: CIM_ServiceAffectsElement

Operation	Requirement	Messages
Associators	Unspecified	None
AssociatorNames	Unspecified	None
References	Unspecified	None
ReferenceNames	Unspecified	None

9 Use Cases (Informative)

This section contains object diagrams and use cases for the *Base Metrics Profile*.

9.1 Instructions Executed per Second

This section contains object diagrams showing several implementations of metrics related to the execution of processor instructions. A management client can use each different type of metric provided to determine the instructions executed per second (IEPS) for the operating system.

9.1.1 Interval Metrics

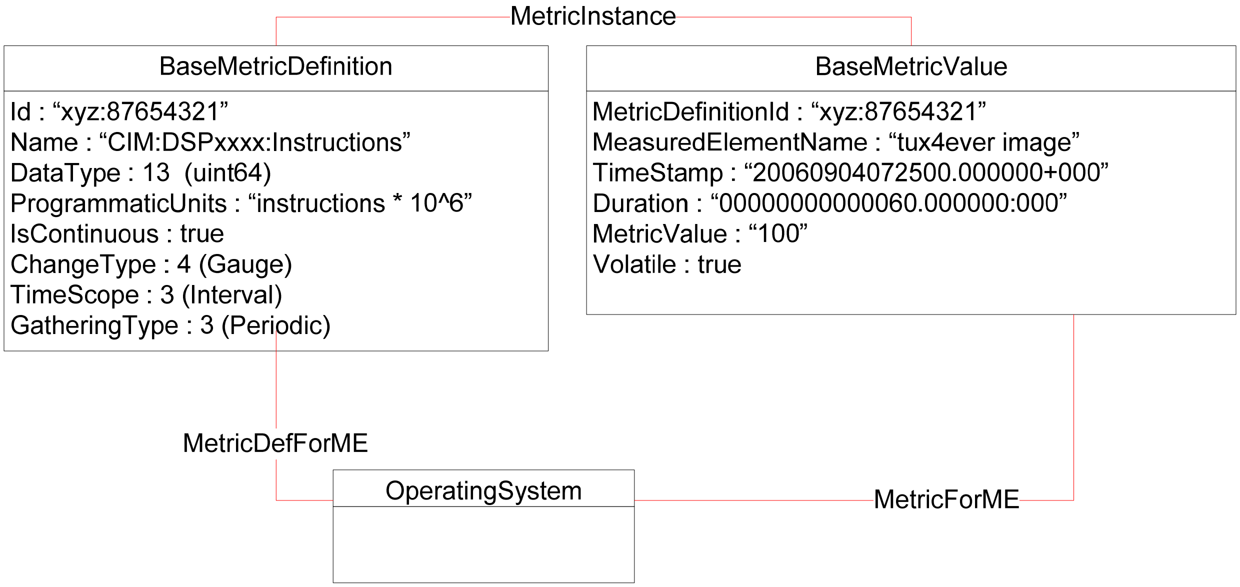
Figure 2 presents an object diagram for an implementation of an interval metric showing the instructions executed per second for an operating system image. There is one instance of the BaseMetricValue class with a TimeStamp property value of 07:25:00 A.M. at 9/4/2006, a Duration property value of 60 seconds and a metric value of 100 million, meaning that the instrumented server has executed 100 million instructions on 9/4/2006 between 07:24:00 A.M. and 07:25:00 A.M. The measured element in this example is an instance of CIM_OperatingSystem. A management client could calculate the average instructions executed per second from 07:24:00 A.M. to 07:25:00 A.M. by dividing the total number of instructions (100 million) by the duration (60 seconds).

The CIM_BaseMetricDefinition.Id property contains a UUID that is chosen by the metrics provider.

The DataType is set to 13 (uint64), which means that the metric values associated to this metric definition instance are intended to be of type uint64.

TimeScope 3 (Interval) means that the metric values are related to a time interval. The values of the TimeStamp and Duration properties indicate that the monitored interval is 09/04/2006 7:24 A.M. UTC through 09/04/2006 7:25 A.M. UTC. The MetricValue property indicates that the operating system has executed 100 million instructions between 7:24:00 A.M. UTC and 7:25:00 A.M. UTC.

GatheringType 3 (Periodic) means that the underlying gathering infrastructure is capturing new counters periodically. How frequently the metric is captured is not indicated. An example would be once a minute.


MetricInstance
BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:Instructions”
DataType : 13 (uint64)
ProgrammaticUnits : “instructions * 10^6”
IsContinuous : true
ChangeType : 4 (Gauge)
TimeScope : 3 (Interval)
GatheringType : 3 (Periodic)
BaseMetricValue
MetricDefinitionId : “xyz:87654321”
MeasuredElementName : “tux4ever image”
TimeStamp : “20060904072500.000000+000”
Duration : “00000000000060.000000:000”
MetricValue : “100”
Volatile : true
MetricDefForME
OperatingSystem
MetricForME

856

857 **Figure 2 – Interval Metrics**

9.1.2 Instantaneous Counter

The object diagram in Figure 3 shows a possible implementation of an instantaneous metric reporting the number of instructions executed. There is exactly one instance of class CIM_BaseMetricValue. The client has executed a GetInstance operation at one minute intervals to query the current values of the metric. The object diagram shows the last retrieved instance using the standard notation. The box with the rounded corners shows the same instance retrieved one minute earlier. A management client can calculate the average IEPS by calculating the delta between the MetricValue properties for the two instances and dividing it by the delta between the TimeStamp properties of the two instances.

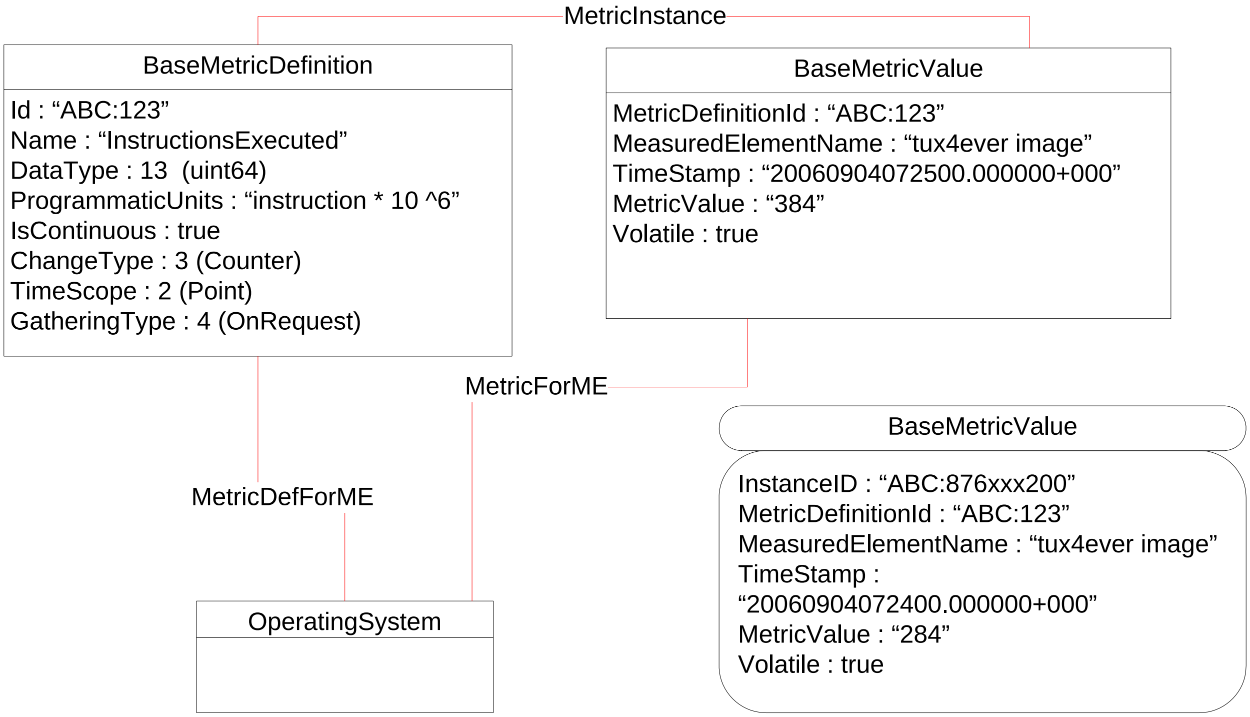
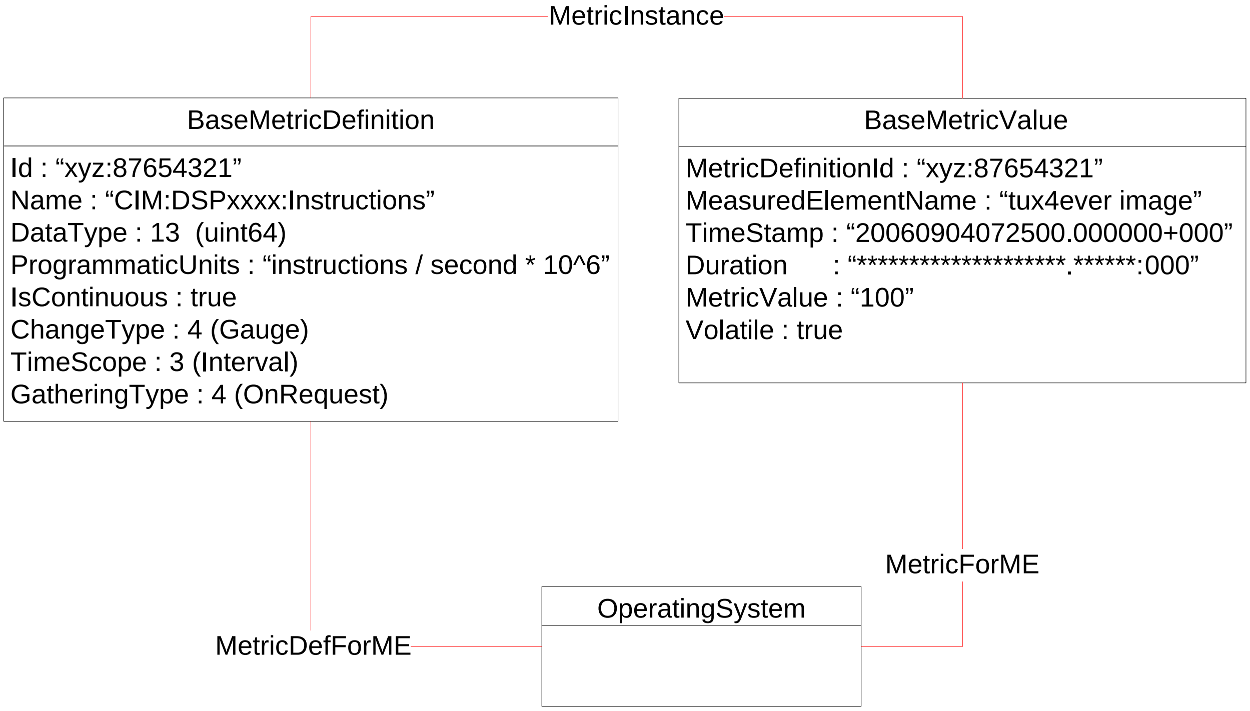


Figure 3 – Instantaneous Counter

9.1.3 Instantaneous Gauge

Figure 4 shows an object diagram in which average instructions per second are directly instrumented. The underlying system provides a metric that corresponds to the average number of instructions per second. However, it does not provide information about the duration over which the average was calculated. This is sometimes known as an instantaneous average.

The ProgrammaticUnits property indicates that the metric reports millions of instructions per second. The CIM_BaseMetricDefinition.TimeScope property indicates that the metric is an interval metric. The CIM_BaseMetricValue.Duration property indicates that there is no precision to the reported interval duration. The current values of the properties of the CIM_BaseMetricValue instance indicate that, as of 07:25:00 A.M. at 9/4/2006, an average of 100 million instructions were executed per second.



BaseMetricDefinition

Id : “xyz:87654321”
Name : “CIM:DSPxxxx:Instructions”
DataType : 13 (uint64)
ProgrammaticUnits : “instructions / second * 10^6”
IsContinuous : true
ChangeType : 4 (Gauge)
TimeScope : 3 (Interval)
GatheringType : 4 (OnRequest)

BaseMetricValue

MetricDefinitionId : “xyz:87654321”
MeasuredElementName : “tux4ever image”
TimeStamp : “20060904072500.000000+000”
Duration : “********************.******:000”
MetricValue : “100”
Volatile : true

Figure 4 – Instantaneous Gauge

9.2 Object Diagram for Startup Interval Time Scope

A value of 4 (StartupInterval) for the TimeScope property indicates that the metric applies to an interval that began at the startup of the measured resource. The example in Figure 5 shows that at 07:25:00 A.M. on 09/04/2006, the associated application system "DB2 V9.1 on tux4ever" was running for a duration of 977 days and 5 hours, consuming 100 million resources. The associated metric is "InstructionsExecuted", with a unit of "Million Instructions".

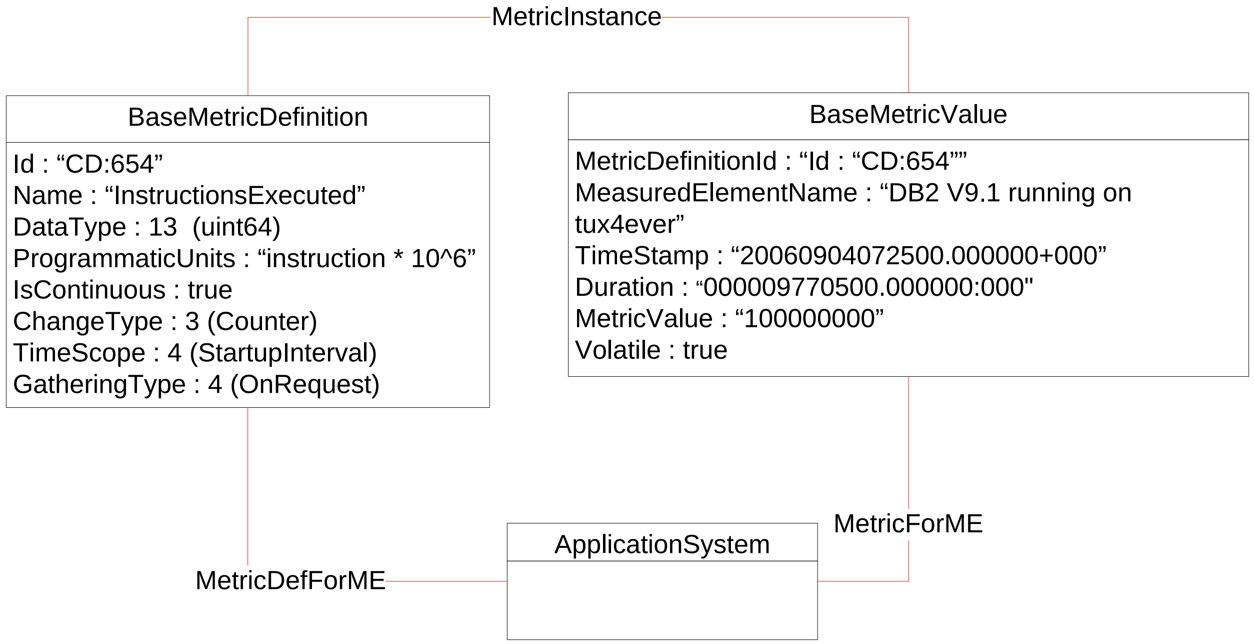

MetricInstance
BaseMetricDefinition
Id : “CD:654”
Name : “InstructionsExecuted”
DataType : 13 (uint64)
ProgrammaticUnits : “instruction * 10^6”
IsContinuous : true
ChangeType : 3 (Counter)
TimeScope : 4 (StartupInterval)
GatheringType : 4 (OnRequest)
BaseMetricValue
MetricDefinitionId : “Id : “CD:654””
MeasuredElementName : “DB2 V9.1 running on tux4ever”
TimeStamp : “20060904072500.000000+000”
Duration : “000009770500.000000:000"
MetricValue : “100000000”
Volatile : true
MetricForME
ApplicationSystem
MetricDefForME

Figure 5 – Usage Example for Startup Interval Time Scope

9.3 Metric Definition for Multiple Instances of CIM_ManagedElement

Figure 6 is an object diagram for an implementation that reports the same metric for two managed elements. metval1 and metval2 report the standard metric "CIM:DSPxxx:InstructionsPerSecond" for cpu1 and cpu2, respectively.

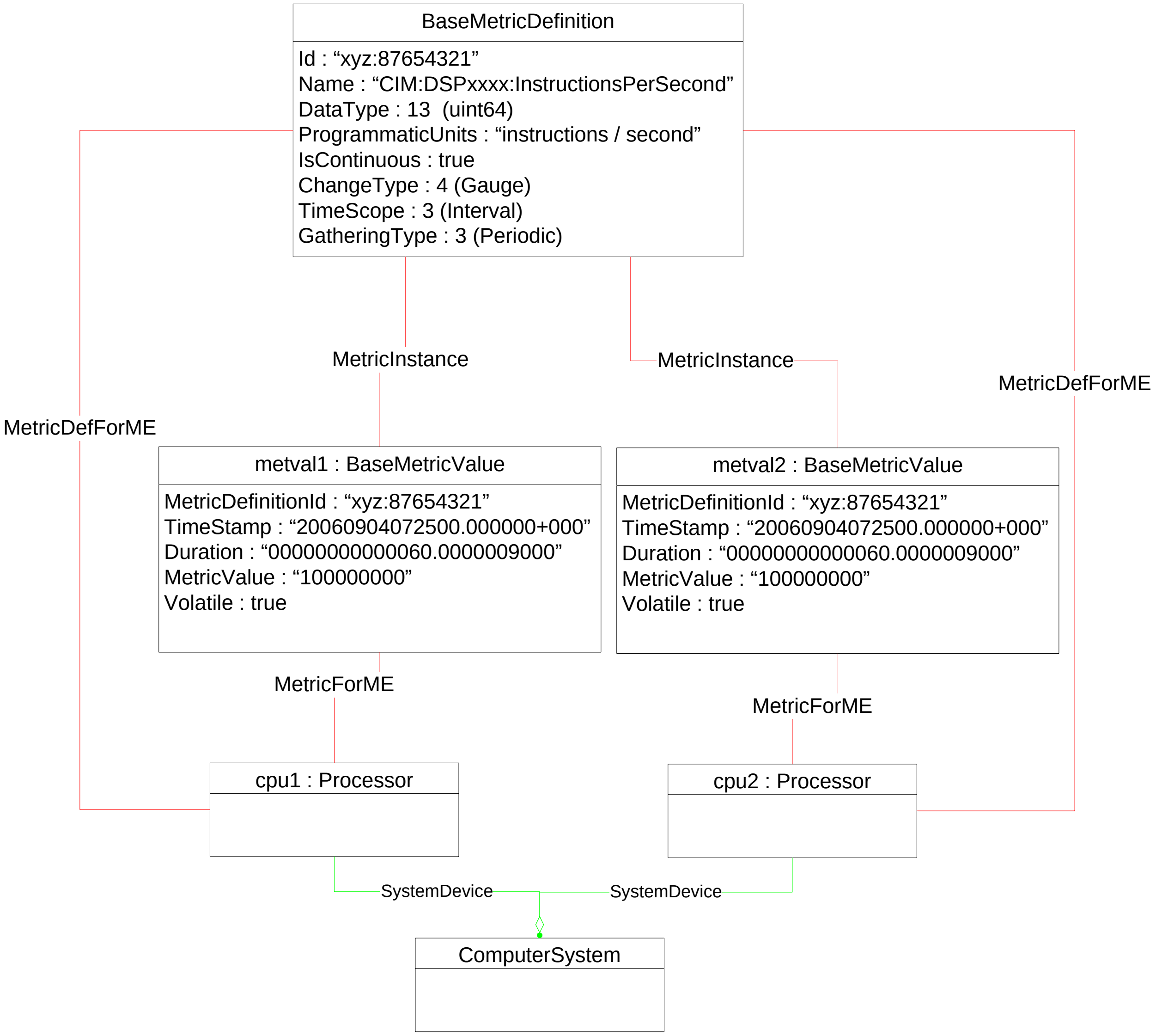

BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
DataType : 13 (uint64)
ProgrammaticUnits : “instructions / second”
IsContinuous : true
ChangeType : 4 (Gauge)
TimeScope : 3 (Interval)
GatheringType : 3 (Periodic)
MetricInstance
MetricInstance
MetricDefForME
MetricDefForME
metval1 : BaseMetricValue
MetricDefinitionId : “xyz:87654321”
TimeStamp : “20060904072500.000000+000”
Duration : “00000000000060.0000009000”
MetricValue : “100000000”
Volatile : true
metval2 : BaseMetricValue
MetricDefinitionId : “xyz:87654321”
TimeStamp : “20060904072500.000000+000”
Duration : “00000000000060.0000009000”
MetricValue : “100000000”
Volatile : true
MetricForME
MetricForME
cpu1 : Processor
cpu2 : Processor
SystemDevice
SystemDevice
ComputerSystem

Figure 6 – Common Metric Definition for Multiple Instances of CIM_ManagedElement

9.4 Controllable Metrics

Figure 7 shows an object diagram for an implementation in which a single metric, represented by the CIM_BaseMetricDefinition instance bmd1, is available for two processors, represented by CIM_Processor instances cpu1 and cpu2. Enabling and disabling the collection of the metric for cpu1 and cpu2 is performed separately. The capabilities for controlling metric collection are indicated by capabilities1. The value of the ControllableMetrics property is bmd1, which indicates that some amount of control over metric collection for values of bmd1 is supported. The value of the MetricControlTypes property is 2 (Discrete), which indicates that metric collection can be controlled for individual values. The value of the ControllableManagedElements property is NULL or empty. The absence of a specific list of CIM_ManagedElement instances associated with bmd1 indicates that controlling metric collection for all metric values of bmd1 is supported. The CIM_ServiceAffectsElement associations between the CIM_MetricService instance and the CIM_BaseMetricDefinition instances have been elided.

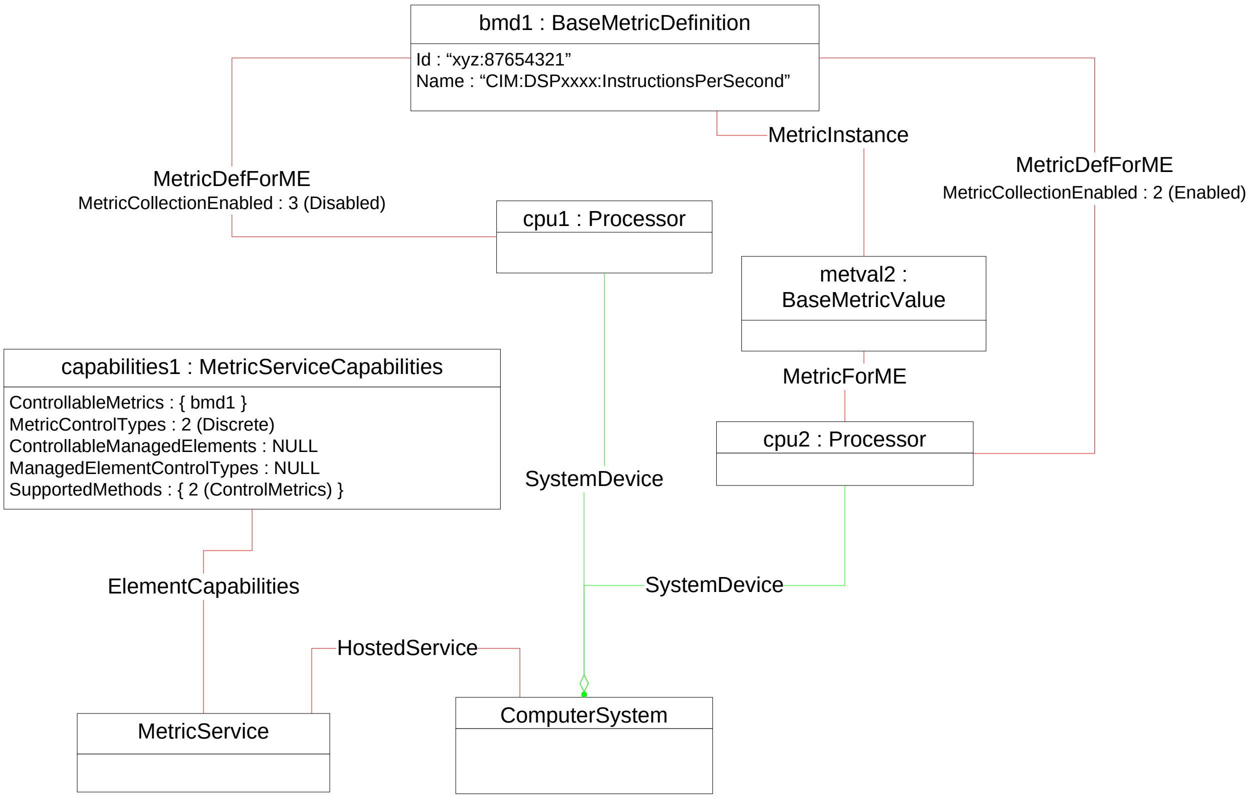

bmd1 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
MetricInstance
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 2 (Enabled)
cpu1 : Processor
metval2 : BaseMetricValue
capabilities1 : MetricServiceCapabilities
ControllableMetrics : { bmd1 }
MetricControlTypes : 2 (Discrete)
ControllableManagedElements : NULL
ManagedElementControlTypes : NULL
SupportedMethods : { 2 (ControlMetrics) }
MetricForME
cpu2 : Processor
SystemDevice
ElementCapabilities
SystemDevice
HostedService
MetricService
ComputerSystem

Figure 7 – Advertising Support for Discrete Controllable Metrics

Figure 8 shows an object diagram for an implementation in which a single metric, represented with the CIM_BaseMetricDefinition instance bmd1, is available for two processors, represented by the CIM_Processor instances cpu1 and cpu2.

The ability to control metrics supported by the implementation shown in Figure 8 is identical to those of the implementation shown in Figure 7. Figure 8 shows an alternate method of advertising the support. The value of the ControllableMetrics property is bmd1, which indicates that some amount of control over metric collection for values of bmd1 is supported. The value of the MetricControlTypes property is 2 (Discrete), which indicates that metric collection can be controlled for individual values. The value of the ControllableManagedElements property is cpu1 and cpu2, which indicates that some amount of control over metrics for cpu1 and cpu2 is supported.

In the object diagram shown in Figure 8, collection of the metric for cpu1 has been disabled. This is indicated by the value of the MetricCollectionEnabled property of the instance of CIM_MetricDefForME that associates bmd1 with cpu1. The CIM_ServiceAffectsElement associations between the CIM_MetricService instance and the CIM_BaseMetricDefinition instances have been elided.

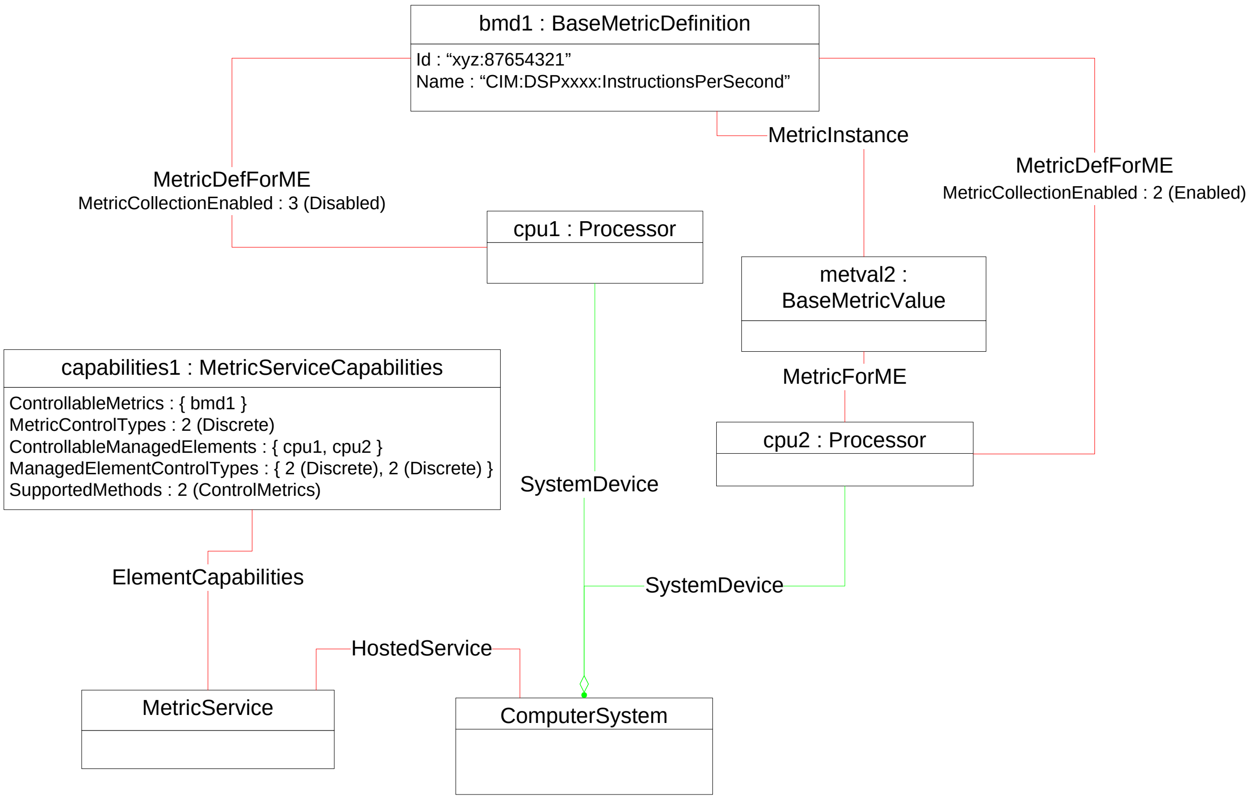
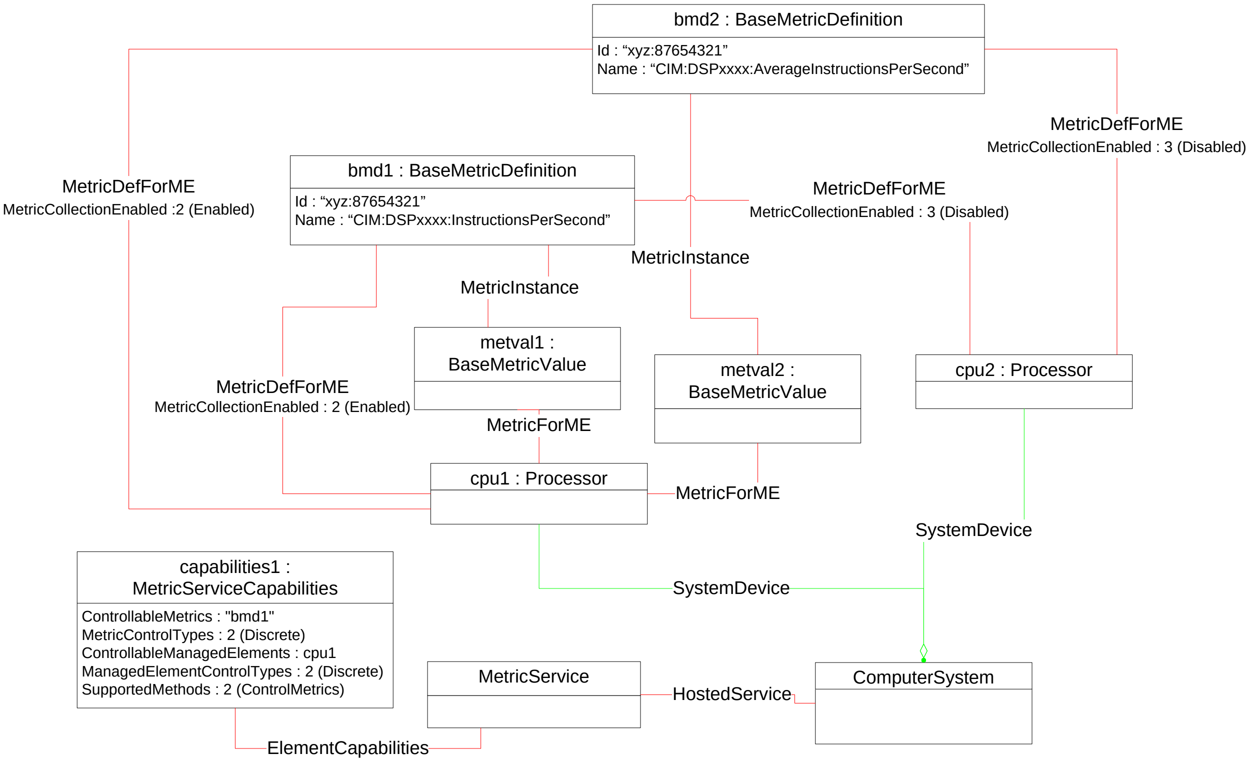

bmd1 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
MetricInstance
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 2 (Enabled)
cpu1 : Processor
metval2 : BaseMetricValue
capabilities1 : MetricServiceCapabilities
ControllableMetrics : { bmd1 }
MetricControlTypes : 2 (Discrete)
ControllableManagedElements : { cpu1, cpu2 }
ManagedElementControlTypes : { 2 (Discrete), 2 (Discrete) }
SupportedMethods : 2 (ControlMetrics)
MetricForME
cpu2 : Processor
SystemDevice
SystemDevice
ElementCapabilities
HostedService
MetricService
ComputerSystem

Figure 8 – Discrete Controllable Metrics (Before Enable)

924 Figure 9 shows an object diagram for the system shown in Figure 8. The
925 CIM_MetricService.ControlMetrics( ) method has been used to enable the collection of the metric
926 represented by the bmd1 instance for cpu1. The CIM_ServiceAffectsElement associations between the
927 CIM_MetricService instance and the CIM_BaseMetricDefinition instances have been elided.

928 
bmd2 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:AverageInstructionsPerSecond”
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled :2 (Enabled)
bmd1 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricInstance
MetricInstance
metval1 : BaseMetricValue
metval2 : BaseMetricValue
cpu2 : Processor
MetricDefForME
MetricCollectionEnabled : 2 (Enabled)
MetricForME
cpu1 : Processor
MetricForME
SystemDevice
capabilities1 : MetricServiceCapabilities
ControllableMetrics : "bmd1"
MetricControlTypes : 2 (Discrete)
ControllableManagedElements : cpu1
ManagedElementControlTypes : 2 (Discrete)
SupportedMethods : 2 (ControlMetrics)
SystemDevice
MetricService
HostedService
ComputerSystem
ElementCapabilities

929 **Figure 9 – Discrete Controllable Metrics (After Enable)**

Figure 10 shows an object diagram for an implementation where two metrics, represented by the CIM_BaseMetricDefinition instances bmd1 and bmd2, are available for two processors, represented by CIM_Processor instances cpu1 and cpu2. The collection of all metric values for the bmd2 instance is controlled as a single operation. The collection of metric values for the bmd1 instance is controlled discretely for each metric value. In the object diagram shown in Figure 10, collection of the metric represented by bmd2 has been disabled. This is indicated by the value of the MetricCollectionEnabled property of the instances of CIM_MetricDefForME that associate bmd2 with cpu1 and cpu2. The CIM_ServiceAffectsElement associations between the CIM_MetricService instance and the CIM_BaseMetricDefinition instances have been elided.

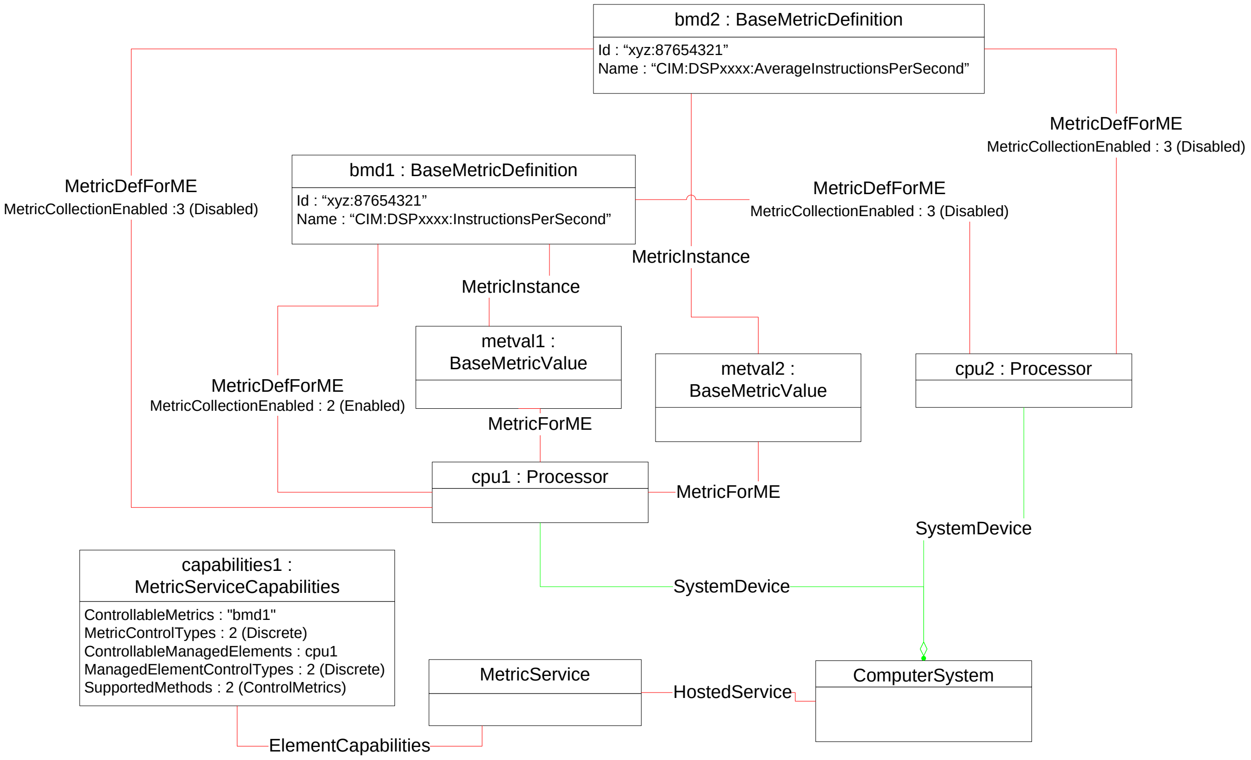

bmd2 : BaseMetricDefinition
Id : "xyz:87654321"
Name : "CIM:DSPxxxx:AverageInstructionsPerSecond"
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
bmd1 : BaseMetricDefinition
Id : "xyz:87654321"
Name : "CIM:DSPxxxx:InstructionsPerSecond"
MetricDefForME
MetricCollectionEnabled :3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricInstance
MetricInstance
metval1 : BaseMetricValue
metval2 : BaseMetricValue
cpu2 : Processor
MetricDefForME
MetricCollectionEnabled : 2 (Enabled)
MetricForME
cpu1 : Processor
MetricForME
SystemDevice
capabilities1 : MetricServiceCapabilities
ControllableMetrics : "bmd1"
MetricControlTypes : 2 (Discrete)
ControllableManagedElements : cpu1
ManagedElementControlTypes : 2 (Discrete)
SupportedMethods : 2 (ControlMetrics)
SystemDevice
MetricService
HostedService
ComputerSystem
ElementCapabilities

Figure 10 – Bulk Controllable Metrics by Definition

Figure 11 shows an object diagram for an implementation in which two metrics, represented by the CIM_BaseMetricDefinition instances bmd1 and bmd2, are available for three processors, represented by CIM_Processor instances cpu1, cpu2, and cpu3. The collection of all metric values for bmd2 is controlled as a single operation. The collection of metric values for bmd1 is controlled discretely for each metric value. In the object diagram shown in Figure 11, collection of metric values for bmd2 has been disabled. This is indicated by the value of the MetricCollectionEnabled property of the instances of CIM_MetricDefForME that associate bmd2 with cpu3 and cpu2. The CIM_ServiceAffectsElement associations between the CIM_MetricService instance and the CIM_BaseMetricDefinition instances have been elided.

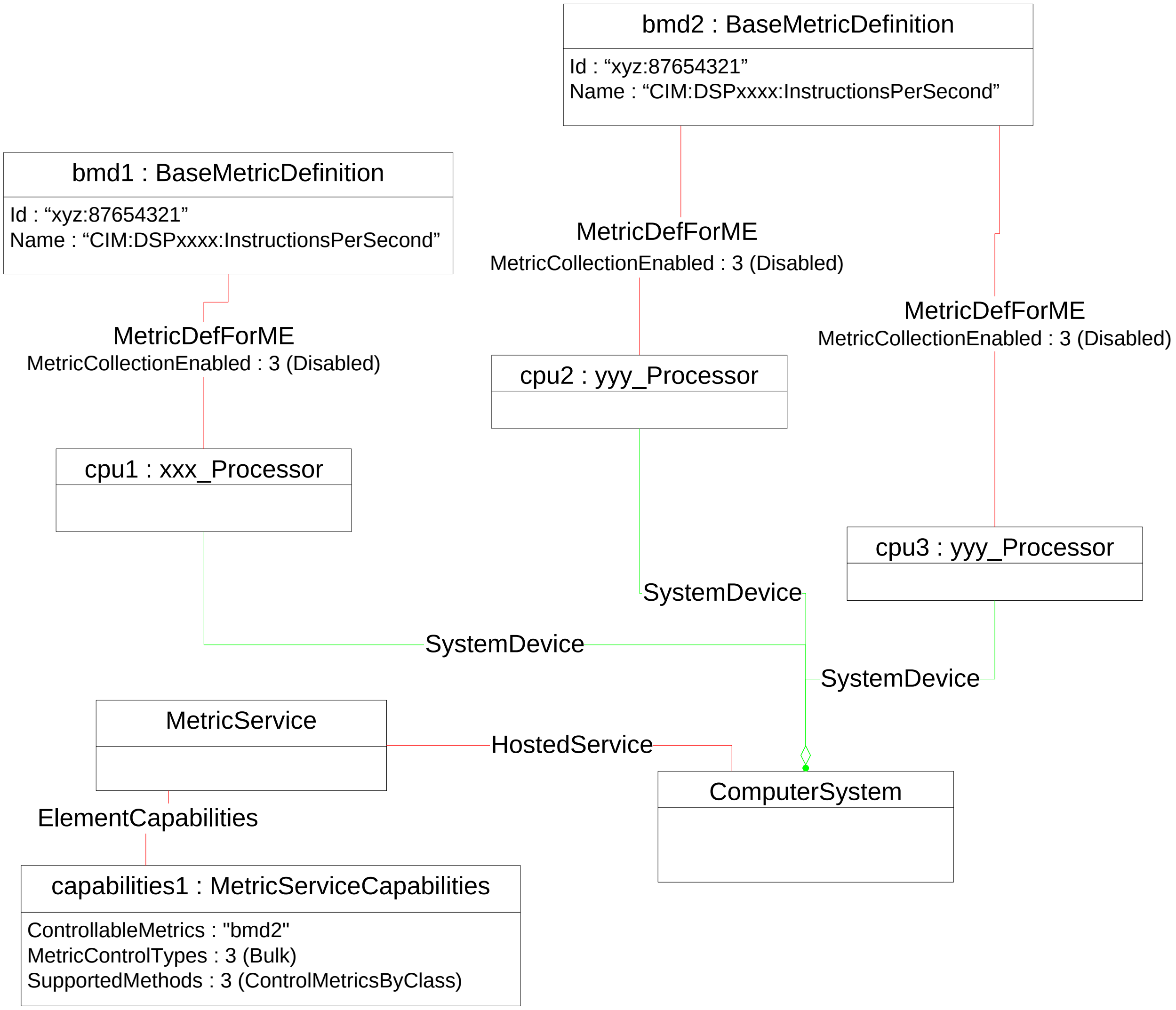
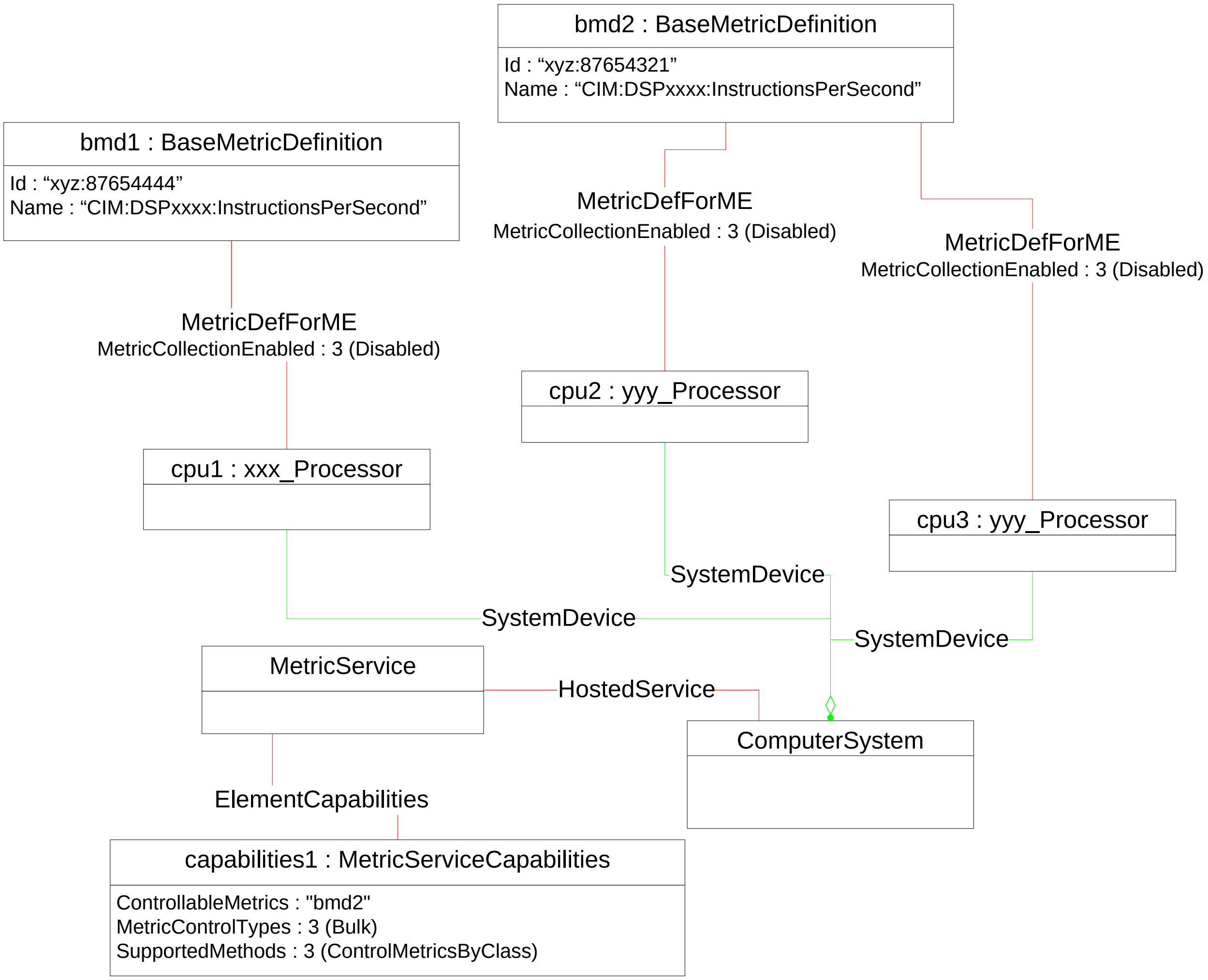

bmd2 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
bmd1 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
cpu2 : yyy_Processor
cpu1 : xxx_Processor
cpu3 : yyy_Processor
SystemDevice
SystemDevice
SystemDevice
MetricService
HostedService
ComputerSystem
ElementCapabilities
capabilities1 : MetricServiceCapabilities
ControllableMetrics : "bmd2"
MetricControlTypes : 3 (Bulk)
SupportedMethods : 3 (ControlMetricsByClass)

Figure 11 – Bulk Controllable Metrics by Managed Element

Figure 12 shows an object diagram for an implementation in which a single metric is available for three processors, represented by instances of subclasses of CIM_Processor cpu1, cpu2, and cpu3. Two instances of CIM_BaseMetricDefinition (bmd1 and bmd2) define the same standard metric "CIM:DSPxxxx"InstructionsPerSection". Multiple instances of the CIM_BaseMetricDefinition class are required in order to represent the separate control points for collection of the metric values. The collection of bmd2 is controlled for all instances of the yyy_Processor class as a bulk operation. Control of the collection of the metric value defined by bmd1 for cpu1 is not supported. The CIM_ServiceAffectsElement

959 associations between the CIM_MetricService instance and the CIM_BaseMetricDefinition instances have
960 been elided.

961


bmd2 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
bmd1 : BaseMetricDefinition
Id : “xyz:87654444”
Name : “CIM:DSPxxxx:InstructionsPerSecond”
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
MetricDefForME
MetricCollectionEnabled : 3 (Disabled)
cpu2 : yyy_Processor
cpu1 : xxx_Processor
cpu3 : yyy_Processor
SystemDevice
SystemDevice
SystemDevice
MetricService
HostedService
ComputerSystem
ElementCapabilities
capabilities1 : MetricServiceCapabilities
ControllableMetrics : "bmd2"
MetricControlTypes : 3 (Bulk)
SupportedMethods : 3 (ControlMetricsByClass)

962 **Figure 12 – Bulk Controllable Metrics by Class**

9.5 Aggregation Metrics

Figure 13 shows an object diagram for an implementation that supports reporting a high watermark for the number of instructions per second executed on a processor. The maximum value in the approximate interval from 12/28/2005 through 11/04/2006 occurred on 09/04/2006 at 7:25 A.M. UTC.

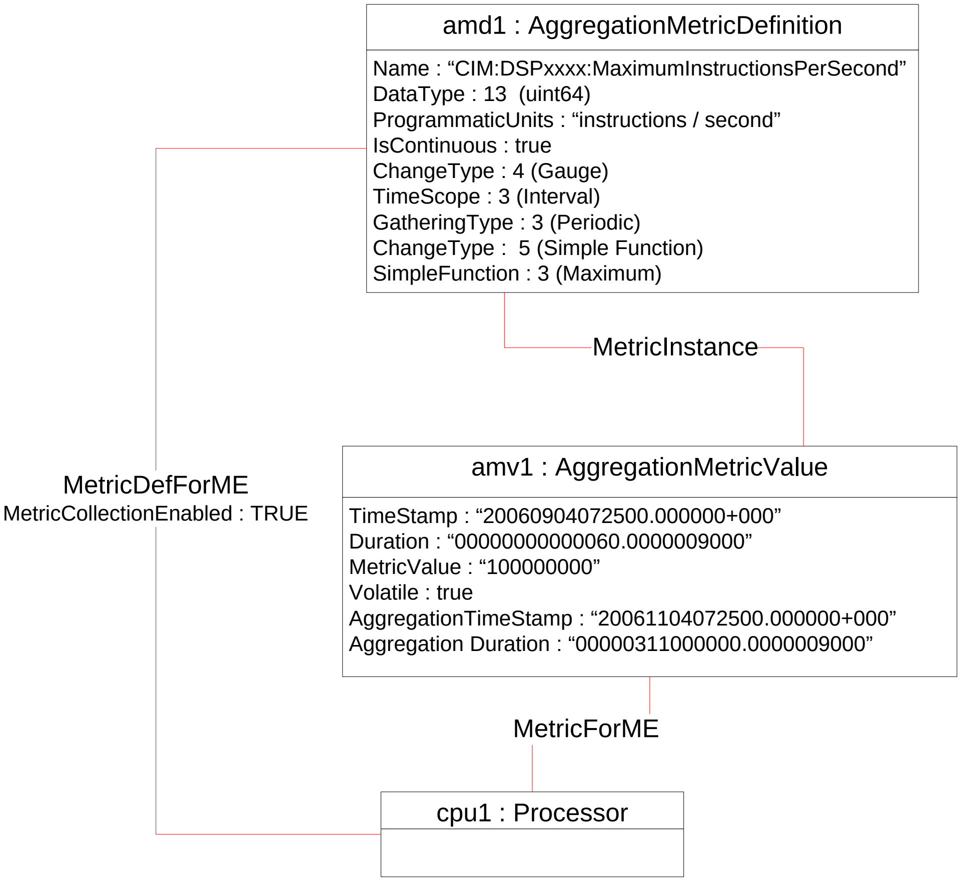

amd1 : AggregationMetricDefinition
Name : “CIM:DSPxxxx:MaximumInstructionsPerSecond”
DataType : 13 (uint64)
ProgrammaticUnits : “instructions / second”
IsContinuous : true
ChangeType : 4 (Gauge)
TimeScope : 3 (Interval)
GatheringType : 3 (Periodic)
ChangeType : 5 (Simple Function)
SimpleFunction : 3 (Maximum)
MetricInstance
MetricDefForME
MetricCollectionEnabled : TRUE
amv1 : AggregationMetricValue
TimeStamp : “20060904072500.000000+000”
Duration : “00000000000060.0000009000”
MetricValue : “100000000”
Volatile : true
AggregationTimeStamp : “20061104072500.000000+000”
Aggregation Duration : “00000311000000.0000009000”
MetricForME
cpu1 : Processor

Figure 13 – Aggregation Metric without Base

Figure 14 shows an object diagram for an implementation that provides the same function as the implementation shown in Figure 13 with the additional functionality of supporting the underlying base metric. The information that bmd1 is the base metric for amd1 is conveyed by the instance of CIM_ConcreteDependency that associates them. In this implementation, long-term monitoring is supported for bmd1; hence, the instance metval1 exists even though it represents historical data.

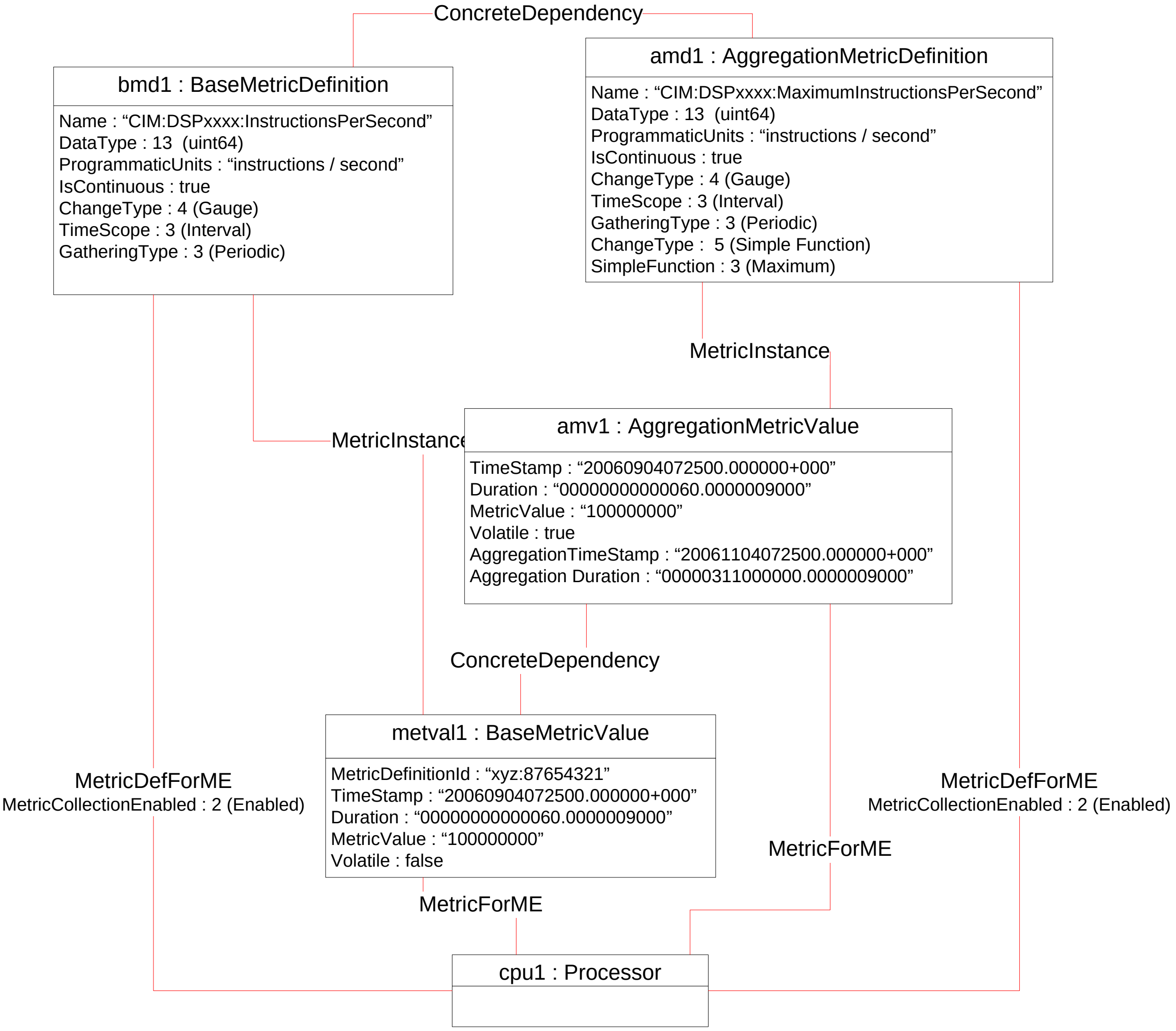

ConcreteDependency
bmd1 : BaseMetricDefinition
Name : “CIM:DSPxxxx:InstructionsPerSecond”
DataType : 13 (uint64)
ProgrammaticUnits : “instructions / second”
IsContinuous : true
ChangeType : 4 (Gauge)
TimeScope : 3 (Interval)
GatheringType : 3 (Periodic)
amd1 : AggregationMetricDefinition
Name : “CIM:DSPxxxx:MaximumInstructionsPerSecond”
DataType : 13 (uint64)
ProgrammaticUnits : “instructions / second”
IsContinuous : true
ChangeType : 4 (Gauge)
TimeScope : 3 (Interval)
GatheringType : 3 (Periodic)
ChangeType : 5 (Simple Function)
SimpleFunction : 3 (Maximum)
MetricInstance
MetricInstance
amv1 : AggregationMetricValue
TimeStamp : “20060904072500.000000+000”
Duration : “00000000000060.0000009000”
MetricValue : “100000000”
Volatile : true
AggregationTimeStamp : “20061104072500.000000+000”
Aggregation Duration : “00000311000000.0000009000”
ConcreteDependency
metval1 : BaseMetricValue
MetricDefinitionId : “xyz:87654321”
TimeStamp : “20060904072500.000000+000”
Duration : “00000000000060.0000009000”
MetricValue : “100000000”
Volatile : false
MetricDefForME
MetricCollectionEnabled : 2 (Enabled)
MetricDefForME
MetricCollectionEnabled : 2 (Enabled)
MetricForME
MetricForME
cpu1 : Processor

Figure 14 – Aggregation Metric with Base

9.6 Metric Context

Figure 15 illustrates the use of breakdown dimensions to differentiate among multiple instances of CIM_BaseMetricValue that provide instances of the same metric definition. metval1 and metval2 are instances of the metric that indicate the instructions per second consumed by process1 from cpu1 and cpu2, respectively. metval3 provides information about the total instructions per second utilized by process1 from all processors.

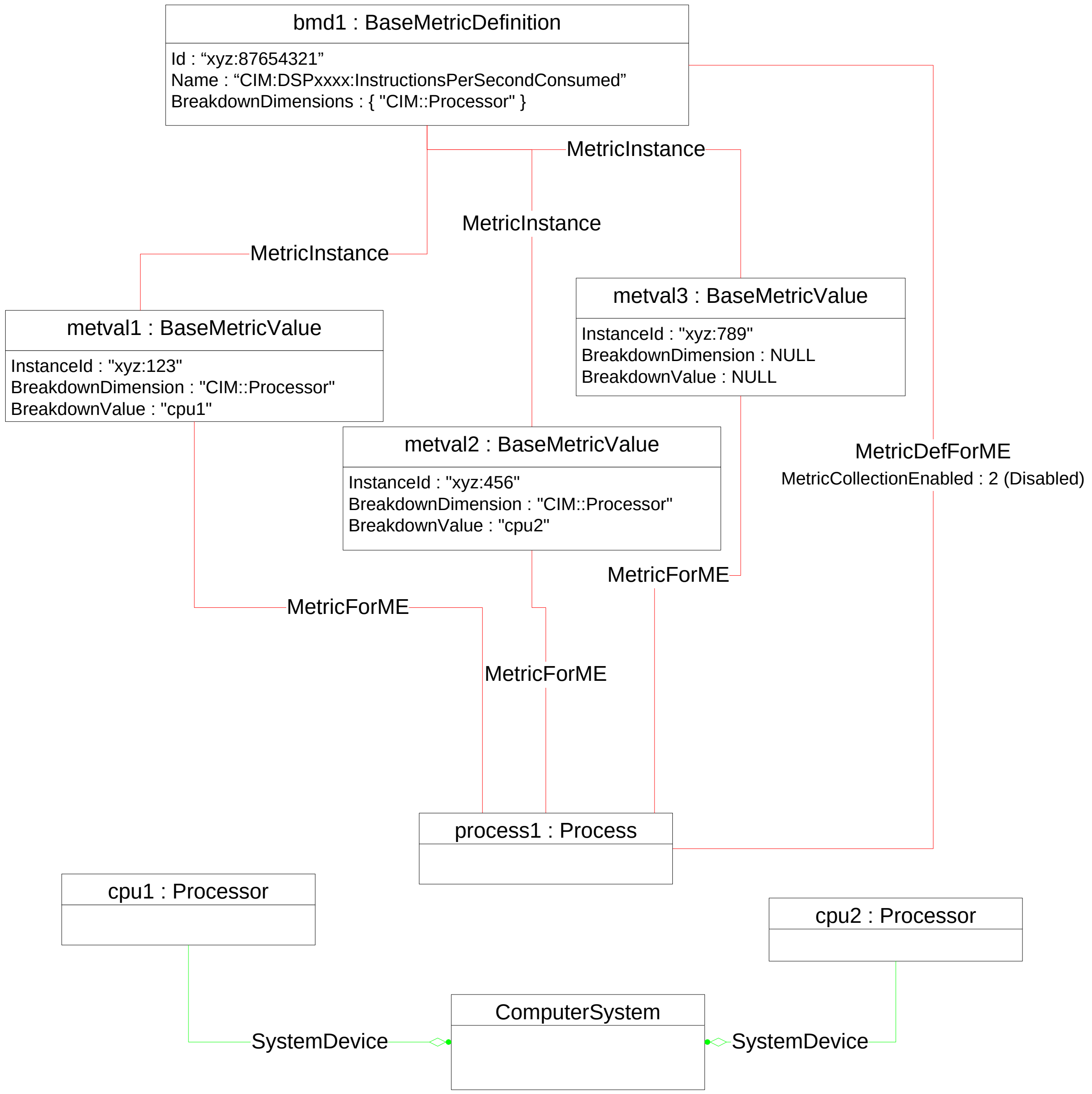

bmd1 : BaseMetricDefinition
Id : “xyz:87654321”
Name : “CIM:DSPxxxx:InstructionsPerSecondConsumed”
BreakdownDimensions : { "CIM::Processor" }
MetricInstance
MetricInstance
MetricInstance
metval3 : BaseMetricValue
InstanceId : "xyz:789"
BreakdownDimension : NULL
BreakdownValue : NULL
metval1 : BaseMetricValue
InstanceId : "xyz:123"
BreakdownDimension : "CIM::Processor"
BreakdownValue : "cpu1"
metval2 : BaseMetricValue
InstanceId : "xyz:456"
BreakdownDimension : "CIM::Processor"
BreakdownValue : "cpu2"
MetricDefForME
MetricCollectionEnabled : 2 (Disabled)
MetricForME
MetricForME
MetricForME
process1 : Process
cpu1 : Processor
cpu2 : Processor
ComputerSystem
SystemDevice
SystemDevice

Figure 15 – Metric Context

9.7 Find All Metric Definitions for a Managed Element

A client can find all of the metric definitions available for a Managed Element as follows:

1) Starting at the instance of CIM_ManagedElement, find all instances of CIM_BaseMetricDefinition associated with the CIM_ManagedElement instance through the CIM_MetricDefForME association.

9.8 Find the Metric Value for a Managed Element

Given an instance of CIM_BaseMetricDefinition that is associated with the CIM_ManagedElement instance through a CIM_MetricDefForME association, a client can find the metric value for the CIM_ManagedElement as follows:

1) Find the instance of CIM_MetricDefForME that associates the CIM_BaseMetricDefinition with the CIM_ManagedElement instance.

2) If the value of the MetricCollectionEnabled property of the instance of CIM_MetricDefForME association found in the previous step is 2 (Enabled):

   a) Find all instances of CIM_BaseMetricValue associated with the CIM_BaseMetricDefinition through the CIM_MetricInstance association.

   b) Find all instances of CIM_BaseMetricValue associated with the CIM_ManagedElement instance through CIM_MetricForME.

   c) Find the instance of CIM_BaseMetricValue that is the intersection of the two result sets by finding matching InstanceID property values.

3) Otherwise, metric collection is disabled and a current data metric value is not available.

9.9 Find a Standard Metric for a Managed Element

Given a string value corresponding to the unique identifier of a standard metric, a client can find the standard metric value for an instance of CIM_ManagedElement as follows:

1) Use the steps in 9.6 to find all metric definitions available for the instance of CIM_ManagedElement.

2) For each instance of CIM_BaseMetricDefinition returned, determine if the Name property matches the string identifier. If there is a match, use the steps in 9.8 to find the metric value.

3) If a matching Name property is not found, the standard metric is not supported for the instance of CIM_ManagedElement.

9.10 Retrieve a Metric Value

A client can retrieve a metric value as follows:

1) Using the steps in 9.9, find the instance of CIM_BaseMetricValue that reports the metric.

2) Invoke GetInstance to query the current values of properties of the CIM_BaseMetricValue instance.

9.11 Find All Metrics Available for a Managed Element within an Enumeration Scope

Given an instance of CIM_ManagedElement, a client can find all of the metrics available for an instance of CIM_ManagedElement as follows:

1) Enumerate all instances of CIM_MetricService within the enumeration scope.

   a) For each instance of CIM_MetricService, find the instance of CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities.

   b) Query the value of the CIM_MetricServiceCapabilities.SupportedMetrics property.

   c) If the array contains the value 4 (Show Metrics), invoke the CIM_MetricService.ShowMetrics( ) method providing the reference to the CIM_ManagedElement.

   d) The list of references to CIM_BaseMetricDefinition returned as the value of the Definitions parameter identifies instances of CIM_BaseMetricDefinition that are available for the CIM_ManagedElement instance.

2) Starting with a reference to the instance of CIM_ManagedElement, find all instances of CIM_MetricDefinition that are associated through CIM_MetricDefForME.

3) Union the results of d) and 2).

9.12 Find All Metrics Available within an Enumeration Scope for All Instances of a CIM Class

Given a CIM class name, a client can find all of the metrics available within an enumeration scope for all instances of the class as follows:

1) Enumerate all instances of CIM_MetricService within the enumeration scope.

   a) For each instance of CIM_MetricService, find the instance of CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities.

   b) Query the value of the CIM_MetricServiceCapabilities.SupportedMetrics property.

   c) If the array contains the value 5 (ShowMetricsByClass), invoke the CIM_MetricService.ShowMetrics( ) method providing the reference to the CIM_ManagedElement.

   d) The list of references to CIM_BaseMetricDefinition returned as the value of the Definitions parameter identifies instances of CIM_BaseMetricDefinition that are available for the CIM_ManagedElement instance.

2) Enumerate all instances of the CIM class.

   a) For each instance of the CIM class, find all instances of CIM_BaseMetricDefinition that are associated through CIM_MetricDefForME.

3) Form a set of instances of CIM_BaseMetricDefinition from the intersection of the instances returned by a).

4) Union the results of d) and 3).

9.13 Determine whether a Metric Can Be Discretely Controlled for a Specific Managed Element

Given an instance of CIM_BaseMetricDefinition associated to an instance of CIM_ManagedElement through CIM_MetricDefForME, a client can determine whether the metric can be controlled for the managed element as follows:

1) Starting with the instance of CIM_BaseMetricDefinition, find the instance of CIM_MetricService associated through CIM_ServiceAffectsElement.

2) Find the instance of CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities with the instance of CIM_MetricService found in 1).

3) If the following conditions are met, the metric can be discretely controlled for the managed element:

   a) The CIM_MetricServiceCapabilities.ControllableMetrics property contains a reference to the CIM_BaseMetricDefinition.

   b) The CIM_MetricServiceCapabilities.MetricControlTypes property contains the value 2 (Discrete) or 4 (Both) at the same array index as the reference in the previous step.

   c) The CIM_MetricServiceCapabilities.ControllableManagedElements property contains a reference to the CIM_ManagedElement.

   d) The CIM_MetricServiceCapabilities.ManagedElementControlTypes property contains the value 2 (Discrete) or 4 (Both) at the same array index as the reference in the previous step.

4) Otherwise, the metric can not be discretely controlled for the managed element.

9.14 Enable a Specific Metric for a Specific Managed Element

Given an instance of CIM_BaseMetricDefinition that is associated to an instance of CIM_ManagedElement through the CIM_MetricDefForME association, a client can enable a specific metric for the managed element as follows:

1) Use the steps in 9.13 to determine whether the metric can be controlled.

2) Invoke the CIM_MetricService.ControlMetrics( ) method and specify the reference to the CIM_ManagedElement as the value of the Subject parameter, the reference to the CIM_BaseMetricDefinition as the value of the Definition parameter, and TRUE as the value of the MetricCollectionEnabled parameter.

9.15 Find All Managed Elements within an Enumeration Scope for which a Metric Is Currently Being Collected

Given an instance of CIM_BaseMetricDefinition, a client can find all instances of CIM_ManagedElement for which the metric is available as follows:

1) Find the instance of CIM_MetricService associated with the CIM_BaseMetricDefinition through the CIM_ServiceAffectsElement association.

2) Find the instance of CIM_MetricServiceCapabilities associated with the CIM_MetricService found in the previous step.

3) Query the value of the SupportedMethods property of the instance of CIM_MetricServiceCapabilities found in the previous step to determine if it contains a value of 4 (ShowMetrics):

   a) If the SupportedMethods property contains a value of 4 (ShowMetrics):

      – Invoke the CIM_MetricService.ShowMetrics( ) method, specifying the reference to the CIM_BaseMetricDefinition as the value of the Definition parameter.

– Upon successful completion of the method, the ManagedElements parameter contains a list of references to CIM_ManagedElement instances for which the metric defined by the CIM_BaseMetricDefinition is available. The MetricCollectionEnabled property indicates whether the metric is currently being collected for the CIM_ManagedElement instance.

4) If the SupportedMethods property does not contain the value 4, find all instances of CIM_MetricDefForME that reference the CIM_BaseMetricDefinition instance. For each instance of CIM_MetricDefForME, the Antecedent property identifies a CIM_ManagedElement for which the metric may be collected and the MetricCollectionEnabled property indicates whether the metric is currently being collected.

10 CIM Elements

Table 19 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 19. Sections 7 ("Implementation") and 8 ("Methods") may impose additional requirements on these elements.

Table 19 – CIM Elements: Base Metrics Profile

Element Name	Requirement	Description
Classes		
CIM_AggregationMetricDefinition	Optional	See 10.1, 10.2, and 10.3.
CIM_AggregationMetricValue	Optional	See 10.4.
CIM_BaseMetricDefinition	Optional	See 10.5, 10.6, 10.7, 10.8, 10.9, and 10.10.
CIM_BaseMetricValue	Optional	See 10.11, 10.12, 10.13, 10.15, and 10.16.
CIM_ConcreteDependency	Optional	See 10.17 and 10.18.
CIM_ElementCapabilities	Mandatory	See 10.19.
CIM_HostedService	Mandatory	See 10.20.
CIM_MetricDefForME	Mandatory	See 10.21.
CIM_MetricForME	Conditional	See 10.22.
CIM_MetricInstance	Conditional	See 10.23.
CIM_MetricService	Mandatory	See 10.24.
CIM_MetricServiceCapabilities	Mandatory	See 10.25.
CIM_RegisteredProfile	Mandatory	See 10.26.
CIM_ServiceAffectsElement	Mandatory	See 10.27.
Indications		
None defined in this profile		

10.1 CIM_AggregationMetricDefinition

CIM_AggregationMetricDefinition defines a metric that can be captured. Table 20 defines the requirements for instances of CIM_AggregationMetricDefinition.

Table 20 – Class: CIM_AggregationMetricDefinition

Properties	Requirement	Notes
BreakdownDimensions	Optional	None
Calculatable	Optional	None
ChangeType	Mandatory	Matches 5 (Simple Function)
DataType	Mandatory	None
ElementName	Mandatory	Pattern (".+")
GatheringType	Mandatory	None
Id	Mandatory	Key
IsContinuous	Optional	None
TimeScope	Optional	None
ProgrammaticUnits	Mandatory	None
Name	Mandatory	See 7.1.3.
SimpleFunction	Mandatory	None

10.2 CIM_AggregationMetricDefinition (Low Watermark)

Table 21 defines the requirements for instances of CIM_AggregationMetricDefinition used to define a low watermark metric.

Table 21 – Class: CIM_AggregationMetricDefinition (Low Watermark)

Properties	Requirement	Notes
BreakdownDimensions	Optional	None
Calculatable	Optional	None
ChangeType	Mandatory	None
DataType	Mandatory	None
ElementName	Mandatory	Pattern (".+")
GatheringType	Mandatory	None
Id	Mandatory	Key
IsContinuous	Optional	None
TimeScope	Optional	None
ProgrammaticUnits	Mandatory	None
Name	Mandatory	See 7.1.3.
SimpleFunction	Mandatory	Matches 2 (Minimum)

10.3 CIM_AggregationMetricDefinition (High Watermark)

Table 22 defines the requirements for instances of CIM_AggregationMetricDefinition used to define high watermark metrics.

Table 22 – Class: CIM_AggregationMetricDefinition (High Watermark)

Properties	Requirement	Notes
BreakdownDimensions	Optional	None
Calculatable	Optional	None
ChangeType	Mandatory	None
DataType	Mandatory	None
ElementName	Mandatory	Pattern (".+")
GatheringType	Mandatory	None
Id	Mandatory	Key
IsContinuous	Optional	None
TimeScope	Optional	None
ProgrammaticUnits	Mandatory	None
Name	Mandatory	See 7.1.3.
SimpleFunction	Mandatory	Matches 3 (Maximum)

10.4 CIM_AggregationMetricValue

CIM_AggregationMetricValue conveys the actual recorded data of a metric that has been maintained. Table 23 describes the requirements for instances of CIM_AggregationMetricValue.

Table 23 – Class: CIM_AggregationMetricValue

Properties	Requirement	Notes
MetricDefinitionId	Mandatory	None
MetricValue	Mandatory	None
Volatile	Mandatory	None
InstanceID	Mandatory	Key
BreakdownDimension	Optional	None
BreakdownValue	Optional	None
AggregationTimeStamp	Mandatory	None
AggregationDuration	Mandatory	None

10.5 CIM_BaseMetricDefinition

CIM_BaseMetricDefinition defines a metric that can be captured. Table 24 defines the requirements for instances of CIM_BaseMetricDefinition.

Table 24 – Class: CIM_BaseMetricDefinition

Properties	Requirement	Notes
BreakdownDimensions	Optional	See 7.1.4.
Calculatable	Optional	None
ChangeType	Mandatory	None
DataType	Mandatory	None
ElementName	Mandatory	Pattern (".+")
GatheringType	Mandatory	See 7.1.5.
Id	Mandatory	Key
IsContinuous	Optional	None
TimeScope	Optional	None
ProgrammaticUnits	Mandatory	None
Name	Mandatory	See 7.1.3.

10.6 CIM_BaseMetricDefinition — Instantaneous Metric

Table 25 describes the requirements for using CIM_BaseMetricDefinition to define an Instantaneous Metric. These constraints are in addition to those specified in 10.5.

Table 25 – Class: CIM_BaseMetricDefinition – Instantaneous Metric

Properties	Requirement	Notes
IsContinuous	Mandatory	Matches TRUE
TimeScope	Mandatory	Matches 2 (Point)

10.7 CIM_BaseMetricDefinition — Interval Metric

Table 26 describes the requirements for using CIM_BaseMetricDefinition to define an Interval Metric. These constraints are in addition to those specified in 10.5.

Table 26 – Class: CIM_BaseMetricDefinition – Interval Metric

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 3 (Interval)

10.8 CIM_BaseMetricDefinition — Startup Interval Metric

Table 27 describes the requirements for using CIM_BaseMetricDefinition to define a Startup Interval Metric. These constraints are in addition to those specified in 10.5.

Table 27 – Class: CIM_BaseMetricDefinition – Startup Interval Metric

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 4 (Startup Interval)

10.9 CIM_BaseMetricDefinition — Summation Metric

Table 28 describes the requirements for using CIM_BaseMetricDefinition to define a Summation Metric. These constraints are in addition to those specified in 10.5.

Table 28 – Class: CIM_BaseMetricDefinition – Summation Metric

Properties	Requirement	Notes
ChangeType	Mandatory	Matches 3 (Counter)
DataType	Mandatory	Matches 4 (real32), 5 (real64), 6 (sint16), 7 (sint32), 8 (sint64), 9 (sint8), 11 (unit16), 12 (uint32), 13 (uint64), or 14 (uint8)

10.10 CIM_BaseMetricDefinition — Current Data

Table 29 describes the requirements for using CIM_BaseMetricDefinition to define the metric to be used with current data. These constraints are in addition to those specified in 10.5.

Table 29 – Class: CIM_BaseMetricDefinition – Current Data

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 2 (Point) or 3 (Interval)

10.11 CIM_BaseMetricValue

CIM_BaseMetricValue conveys the actual recorded data of a metric. Table 30 describes the requirements for instances of CIM_BaseMetricValue.

Table 30 – Class: CIM_BaseMetricValue

Properties	Requirement	Notes
MetricDefinitionId	Mandatory	None
MetricValue	Mandatory	None
Volatile	Mandatory	None
InstanceID	Mandatory	Key
BreakdownDimension	Optional	See 7.1.4.
BreakdownValue	Optional	See 7.1.4.
Timestamp	Optional	None
Duration	Optional	None

10.12 CIM_BaseMetricValue — Current Data

CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 31 describes the requirements for using CIM_BaseMetricValue to report the metric for current data. These constraints are in addition to those specified in 10.11.

Table 31 – Class: CIM_BaseMetricValue – Current Data

Properties	Requirement	Notes
Timestamp	Mandatory	None
Volatile	Mandatory	Matches TRUE

10.13 CIM_BaseMetricValue — Interval Metrics

CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 32 describes the requirements for using CIM_BaseMetricValue to report the metric for interval metrics. These constraints are in addition to those specified in 10.11.

Table 32 – Class: CIM_BaseMetricValue – Interval Metrics

Properties	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None

10.14 CIM_BaseMetricValue — Startup Interval Metrics

CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 33 describes the requirements for using CIM_BaseMetricValue to report the metric for startup interval metrics. These constraints are in addition to those specified in 10.11.

Table 33 – Class: CIM_BaseMetricValue – Startup Interval Metrics

Properties	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None

10.15 CIM_BaseMetricValue — Summation Metric

CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 34 describes the requirements for using CIM_BaseMetricValue to report the metric for a Summation Metric. These constraints are in addition to those specified in 10.11.

Table 34 – Class: CIM_BaseMetricValue – Summation Metric

Properties	Requirement	Notes
Timestamp	Mandatory	None

10.16 CIM_BaseMetricValue — Long-Term Monitoring

CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 35 describes the requirements for using CIM_BaseMetricValue to report a metric for long-term monitoring. These constraints are in addition to those specified in 10.11.

Table 35 – Class: CIM_BaseMetricValue – Long-Term Monitoring

Properties	Requirement	Notes
Volatile	Mandatory	Matches FALSE

10.17 CIM_ConcreteDependency (Definition)

Table 36 details the requirements for instances of CIM_ConcreteDependency.

Table 36 – Class: CIM_ConcreteDependency (Definition)

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall be a reference to CIM_BaseMetricDefinition. Cardinality 0..1
Dependent	Mandatory	**Key:** This property shall be a reference to CIM_AggregationMetricDefinition. Cardinality 0..1

10.18 CIM_ConcreteDependency (Value)

Table 37 details the requirements for instances of CIM_ConcreteDependency.

Table 37 – Class: CIM_ConcreteDependency (Value)

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall be a reference to CIM_BaseMetricValue. Cardinality 0..1
Dependent	Mandatory	**Key:** This property shall be a reference to CIM_AggregationMetricValue. Cardinality 0..1

10.19 CIM_ElementCapabilities

CIM_ElementCapabilities associates an instance of CIM_MetricServiceCapabilities with the Central Instance. Table 38 details the requirements for instances of CIM_ElementCapabilities.

Table 38 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
ManagedElement	Mandatory	**Key:** This property shall be a reference to the Central Instance. Cardinality 1
Capabilities	Mandatory	**Key:** This property shall be a reference to an instance of CIM_MetricServiceCapabilities. Cardinality 1

10.20 CIM_HostedService

Table 39 details the requirements for instances of CIM_HostedService.

Table 39 – Class: CIM_HostedService

Elements	Requirement	Notes
Antecedent	Mandatory	**Key:** This property shall be a reference to the Scoping Instance. Cardinality 1
Dependent	Mandatory	**Key:** This property shall be a reference to the Central Instance. Cardinality 1..*

10.21 CIM_MetricDefForME

CIM_MetricForME relates a metric to the managed element for which it was measured. Table 40 details the requirements for instances of CIM_MetricDefForME.

Table 40 – Class: CIM_MetricDefForME

Properties	Requirement	Notes
Antecedent	Mandatory	Cardinality 1..*
Dependent	Mandatory	Cardinality *
MetricCollectionEnabled	Mandatory	None

10.22 CIM_MetricForME

CIM_MetricForME relates a metric to the managed element for which it was measured. Table 41 details the requirements for instances of CIM_MetricForME.

Table 41 – Class: CIM_MetricForME

Properties	Requirement	Notes
Antecedent	Mandatory	Cardinality 1..*
Dependent	Mandatory	Cardinality *

10.23 CIM_MetricInstance

CIM_MetricInstance relates a CIM_BaseMetricValue to the CIM_BaseMetricDefinition that defines it. Table 42 details the requirements for instances of CIM_MetricInstance.

Table 42 – Class: CIM_MetricInstance

Properties	Requirement	Notes
Antecedent	Mandatory	See 7.1.2. Cardinality 1
Dependent	Mandatory	See 7.1.2. Cardinality *

10.24 CIM_MetricService

Table 43 details the requirements for instances of CIM_MetricService.

Table 43 – Class: CIM_MetricService

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	Key
CreationClassName	Mandatory	Key
SystemName	Mandatory	Key
Name	Mandatory	Key
ElementName	Mandatory	Pattern ".*"
ShowMetrics( )	Conditional	See 8.1.
ShowMetricsByClass( )	Conditional	See 8.2.
ControlMetrics( )	Conditional	See 8.3.
ControlMetricsByClass( )	Conditional	See 8.4.
GetMetricValues( )	Conditional	See 8.5.

10.25 CIM_MetricServiceCapabilities

CIM_MetricServiceCapabilities indicates support for managing the state of the service as well as the accounts with which the service is associated. Table 44 details the requirements for instances of CIM_MetricServiceCapabilities.

Table 44 – Class: CIM_MetricServiceCapabilities

Elements	Requirement	Notes
InstanceID	Mandatory	None
ElementName	Mandatory	Pattern ".*"
SupportedMethods	Mandatory	None
ControllableMetrics	Mandatory	None
MetricControlTypes	Mandatory	None
ControllableManagedElements	Mandatory	None
ManagedElementControlTypes	Mandatory	None

10.26 CIM_RegisteredProfile

CIM_RegisteredProfile identifies the *Base Metrics Profile*. The CIM_RegisteredProfile class is defined by the *Profile Registration Profile*. With the exception of the mandatory values specified for the properties in Table 45, the behavior of the CIM_RegisteredProfile instance is in accordance with the constraints specified in the *Profile Registration Profile*.

Table 45 – Class: CIM_RegisteredProfile

Properties	Requirement	Notes
RegisteredName	Mandatory	This property shall have a value of "Base Metrics".
RegisteredVersion	Mandatory	This property shall have a value of "1.0.0".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

10.27 CIM_ServiceAffectsElement

CIM_ServiceAffectsElement is used to associate an instance of CIM_MetricService with an instance of CIM_BaseMetricDefinition or CIM_AggregationMetricDefinition that represents a metric that could be controlled using the service. Table 46 contains the requirements for elements of this class.

Table 46 – Class: CIM_ServiceAffectsElement

Elements	Requirement	Notes
AffectedElement	Mandatory	**Key**: This property shall reference the instance of CIM_BaseMetricDefinition or CIM_AggregationMetricDefinition. Cardinality 1..*
AffectingElement	Mandatory	**Key**: This property shall reference the instance of CIM_MetricService. Cardinality 1
ElementAffects	Mandatory	Matches 5 (Manages)

ANNEX A
(Informative)

Change Log

Version	Date	Author	Description
1.0.0	2009-06-16	Aaron Merkin	DMTF Standard

ANNEX B
(Informative)

Guide for Common Metrics

This annex provides an informative list of the combined mandatory properties for instances of CIM_BaseMetricDefinition, CIM_AggregationMetricDefinition, CIM_BaseMetricValue, and CIM_AggregationMetricValue if used to represent common metrics. Each of the data cells of the tables lists mandatory properties and their value formulations for a specific type of metric. Each table corresponds to a different type of metrics grouped by value formulation. The rows represent the different type of metrics based on the time scope that metric describes. The columns describe the different type of metrics based on the metric collection access type used.

In order to determine the mandatory set of properties, match the type of metric to one of the data cells based on the metric's value formulation, time scope, and collection access type.

The following conventions are used in the table:

- BMD – the properties that follow are required on the instance of CIM_BaseMetricDefinition
- BMV – the properties that follow are required on the instance of CIM_BaseMetricValue
- AMD – the propertiest that follow are required on the instance of CIM_AggregationMetricDefinition
- AMV – the properties that follow are required on the instance of CIM_AggregationMetricValue
- A property name without a value specified is required, and the value is not fixed
- A property name followed by a value assignment is required with the specified value fixed.

NOTE: If there is a mismatch between the mandatory set of properties and/or the properties' value formulation indicated by the tables in this annex and the requirements detailed in 7 and 10, the requirements mandated in 7 and 10 take precedence.

B.1 Simple Metric

Table B-1 describes the mandatory properties for simple metric as described in 6.3.1 according to the appropriate metric access type and time scope.

Table B-1 – Simple Metric

	Current Data	Current Data - Online Monitoring	Current Data - Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Instantaneous Metrics	**BMD** ChangeType DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType = 4 (OnRequest) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name
	BMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID	**BMV** MetricDefinitionId MetricValue Volatile InstanceID
Interval Metrics	**BMD** ChangeType DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id ITimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name
	BMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId Volatile = TRUE Volatile InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp

	Current Data	**Current Data - Online Monitoring**	**Current Data - Snapshot Monitoring**	**Long-Term Monitoring**	**Event-Based Monitoring**
Startup Metrics	**BMD** ChangeType DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name
	BMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp	**BMV** MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp

B.2 Summation Metric

Table B-2 describes the mandatory properties for summation metric as described in 6.3.2 according to the appropriate metric access type and time scope.

Table B-2 – Summation Metric

	Current Data	Current Data - Online Monitoring	Current Data - Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Instantaneous Metrics	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType = 4 (OnRequest) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name
	BMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID	**BMV** MetricDefinitionId MetricValue Volatile InstanceID

	Current Data	Current Data - Online Monitoring	Current Data - Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Interval Metrics	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType = 4 (OnRequest) Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name
	BMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId Volatile = TRUE Volatile InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile InstanceID Timestamp
Startup Metrics	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name	**BMD** ChangeType = 3 (Counter) DataType = 4 (real32) or 5 (real64) or 6 (sint16) or 7 (sint32) or 8 (sint64) or 9 (sint8) or 11 (unit16) or 12 (uint32) or 13 (uint64) or 14 (uint8) ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name

	Current Data	Current Data - Online Monitoring	Current Data - Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
	BMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Timestamp	**BMV** MetricDefinitionId MetricValue Volatile InstanceID Timestamp

B.3 Aggregation Metric

Table B-3 describes the mandatory properties for aggregation metric as described in section 6.3.3 according to the appropriate metric access type and time scope.

Table B-3 – Aggregation Metric

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Instantaneous Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID AggregationTimeStamp AggregationDuration

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Interval Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id ITimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId Volatile = TRUE Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration
Startup Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
	AMV	**AMV**	**AMV**	**AMV**	**AMV**
	MetricDefinitionId	MetricDefinitionId	MetricDefinitionId	MetricDefinitionId	MetricDefinitionId
	MetricValue	MetricValue	MetricValue	MetricValue	MetricValue
	Volatile = TRUE	Volatile = TRUE	Volatile = TRUE	Volatile = FALSE	Volatile
	InstanceID	InstanceID	InstanceID	InstanceID	InstanceID
	Duration	Duration	Duration	Duration	Duration
	Timestamp	Timestamp	Timestamp	Timestamp	Timestamp
	AggregationTimeStamp	AggregationTimeStamp	AggregationTimeStamp	AggregationTimeStamp	AggregationTimeStamp
	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration

B.4 Aggregation Metric — Low Watermark

Table B-4 describes the mandatory properties for low watermark as a type of an aggregation metric as described in section 6.3.3.1 according to the appropriate metric access type and time scope.

Table B-4 – Aggregation Metric – Low Watermark

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Instantaneous Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID AggregationTimeStamp AggregationDuration
Interval Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id ITimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId Volatile = TRUE Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Startup Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration

B.5 Aggregation Metric — High Watermark

Table B-5 describes the mandatory properties for high watermark as a type of an aggregation metric as described in section 6.3.3.1 according to the appropriate metric access type and time scope.

Table B-5 – Aggregation Metric – High Watermark

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Instantaneous Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id IsContinuous = TRUE TimeScope = 2 (Point) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID AggregationTimeStamp AggregationDuration

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Interval Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id ITimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)
	AMV MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId Volatile = TRUE Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration	**AMV** MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeStamp AggregationDuration
Startup Metrics	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)	**AMD** ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 3 (Maximum)

	Current Data	**Current Data – Online Monitoring**	**Current Data – Snapshot Monitoring**	**Long-Term Monitoring**	**Event-Based Monitoring**
	AMV	**AMV**	**AMV**	**AMV**	**AMV**
	MetricDefinitionId	MetricDefinitionId	MetricDefinitionId	MetricDefinitionId	MetricDefinitionId
	MetricValue	MetricValue	MetricValue	MetricValue	MetricValue
	Volatile = TRUE	Volatile = TRUE	Volatile = TRUE	Volatile = FALSE	Volatile
	InstanceID	InstanceID	InstanceID	InstanceID	InstanceID
	Duration	Duration	Duration	Duration	Duration
	Timestamp	Timestamp	Timestamp	Timestamp	Timestamp
	AggregationTimeStamp	AggregationTimeStamp	AggregationTimeStamp	AggregationTimeStamp	AggregationTimeStamp
	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration

Bibliography

DMTF DSP1073, *Capacity Metrics Profile 1.0* (under development)