



1

2

3

4

Document Number: DSP1053

Date: 2009-12-11

Version: 1.0.1

5 **Base Metrics Profile**

6 **Document Type: Specification**

7 **Document Status: DMTF Standard**

8 **Document Language: E**

9

10 Copyright notice

11 Copyright © 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

33	Foreword	7
34	Introduction	8
35	1 Scope	9
36	2 Normative References.....	9
37	3 Terms and Definitions	9
38	4 Symbols and Abbreviated Terms	12
39	5 Synopsis.....	12
40	6 Description (Informative).....	12
41	6.1 Metric Access Types.....	13
42	6.2 Metric Time Scope.....	14
43	6.3 Metric Value Formulation.....	15
44	6.4 Metric Context.....	16
45	7 Implementation.....	16
46	7.1 Common Requirements.....	16
47	7.2 Modeling Metric Access Types.....	18
48	7.3 Modeling Metric Time Scope	18
49	7.4 Modeling Metric Value Formulation	19
50	7.5 Relationship between Aggregation and Base Metrics.....	19
51	7.6 Constraints on Metric Values for Controllable Metrics.....	19
52	8 Methods.....	20
53	8.1 CIM_MetricService.ShowMetrics()	20
54	8.2 CIM_MetricService.ShowMetricsByClass()	21
55	8.3 CIM_MetricService.ControlMetrics().....	22
56	8.4 CIM_MetricService.ControlMetricsByClass()	24
57	8.5 CIM_MetricService.GetMetricValues()	25
58	8.6 Profile Conventions for Operations.....	25
59	8.7 CIM_AggregationMetricDefinition	26
60	8.8 CIM_AggregationMetricValue.....	26
61	8.9 CIM_BaseMetricDefinition	26
62	8.10 CIM_BaseMetricValue	26
63	8.11 CIM_ConcreteDependency	26
64	8.12 CIM_ElementCapabilities	26
65	8.13 CIM_HostedService	27
66	8.14 CIM_MetricDefForME	27
67	8.15 CIM_MetricForME.....	28
68	8.16 CIM_MetricInstance	28
69	8.17 CIM_MetricService.....	28
70	8.18 CIM_MetricServiceCapabilities.....	28
71	8.19 CIM_ServiceAffectsElement	29
72	9 Use Cases (Informative).....	29
73	9.1 Instructions Executed per Second.....	29
74	9.2 Object Diagram for Startup Interval Time Scope.....	33
75	9.3 Metric Definition for Multiple Instances of CIM_ManagedElement.....	34
76	9.4 Controllable Metrics	35
77	9.5 Aggregation Metrics	41
78	9.6 Metric Context.....	43
79	9.7 Find All Metric Definitions for a Managed Element.....	44
80	9.8 Find the Metric Value for a Managed Element	44
81	9.9 Find a Standard Metric for a Managed Element.....	44
82	9.10 Retrieve a Metric Value.....	44
83	9.11 Find All Metrics Available for a Managed Element within an Enumeration Scope.....	45

84	9.12	Find All Metrics Available within an Enumeration Scope for All Instances of a CIM Class	45
85	9.13	Determine whether a Metric Can Be Discretely Controlled for a Specific Managed	
86		Element	46
87	9.14	Enable a Specific Metric for a Specific Managed Element	46
88	9.15	Find All Managed Elements within an Enumeration Scope for which a Metric Is Currently	
89		Being Collected	46
90	10	CIM Elements	47
91	10.1	CIM_AggregationMetricDefinition	47
92	10.2	CIM_AggregationMetricDefinition (Low Watermark)	48
93	10.3	CIM_AggregationMetricDefinition (High Watermark)	48
94	10.4	CIM_AggregationMetricValue	49
95	10.5	CIM_BaseMetricDefinition	49
96	10.6	CIM_BaseMetricDefinition — Instantaneous Metric	50
97	10.7	CIM_BaseMetricDefinition — Interval Metric	50
98	10.8	CIM_BaseMetricDefinition — Startup Interval Metric	50
99	10.9	CIM_BaseMetricDefinition — Summation Metric	50
100	10.10	CIM_BaseMetricDefinition — Current Data	51
101	10.11	CIM_BaseMetricValue	51
102	10.12	CIM_BaseMetricValue — Current Data	51
103	10.13	CIM_BaseMetricValue — Interval Metrics	52
104	10.14	CIM_BaseMetricValue — Startup Interval Metrics	52
105	10.15	CIM_BaseMetricValue — Summation Metric	52
106	10.16	CIM_BaseMetricValue — Long-Term Monitoring	52
107	10.17	CIM_ConcreteDependency (Definition)	53
108	10.18	CIM_ConcreteDependency (Value)	53
109	10.19	CIM_ElementCapabilities	53
110	10.20	CIM_HostedService	54
111	10.21	CIM_MetricDefForME	54
112	10.22	CIM_MetricForME	54
113	10.23	CIM_MetricInstance	54
114	10.24	CIM_MetricService	55
115	10.25	CIM_MetricServiceCapabilities	55
116	10.26	CIM_RegisteredProfile	56
117	10.27	CIM_ServiceAffectsElement	56
118	ANNEX A (Informative)	Change Log	57
119	ANNEX B (Informative)	Guide for Common Metrics	58
120	Bibliography		71
121			
122	Figures		
123	Figure 1 – Base Metrics Profile: Class Diagram		13
124	Figure 2 – Interval Metrics		30
125	Figure 3 – Instantaneous Counter		31
126	Figure 4 – Instantaneous Gauge		32
127	Figure 5 – Usage Example for Startup Interval Time Scope		33
128	Figure 6 – Common Metric Definition for Multiple Instances of CIM_ManagedElement		34
129	Figure 7 – Advertising Support for Discrete Controllable Metrics		35
130	Figure 8 – Discrete Controllable Metrics (Before Enable)		36
131	Figure 9 – Discrete Controllable Metrics (After Enable)		37
132	Figure 10 – Bulk Controllable Metrics by Definition		38
133	Figure 11 – Bulk Controllable Metrics by Managed Element		39
134	Figure 12 – Bulk Controllable Metrics by Class		40
135	Figure 13 – Aggregation Metric without Base		41

136 Figure 14 – Aggregation Metric with Base 42

137 Figure 15 – Metric Context..... 43

138

139 **Tables**

140 Table 1 – Referenced Profiles 12

141 Table 2 – CIM_MetricService.ShowMetrics() Method: Return Code Values 20

142 Table 3 – CIM_MetricService.ShowMetrics() Method: Parameters..... 21

143 Table 4 – CIM_MetricService.ShowMetricsByClass() Method: Return Code Values 21

144 Table 5 – CIM_MetricService.ShowMetricsByClass() Method: Parameters 22

145 Table 6 – CIM_MetricService.ControlMetrics() Method: Return Code Values 22

146 Table 7 – CIM_MetricService.ControlMetrics() Method: Parameters 23

147 Table 8 – CIM_MetricService.ControlMetricsByClass() Method: Return Code Values 24

148 Table 9 – CIM_MetricService.ControlMetricsByClass() Method: Parameters..... 24

149 Table 10 – CIM_MetricService.GetMetricValues() Method: Return Code Values 25

150 Table 11 – CIM_MetricService.GetMetricValues() Method: Parameters..... 25

151 Table 12 – Operations: CIM_ConcreteDependency 26

152 Table 13 – Operations: CIM_ElementCapabilities 27

153 Table 14 – Operations: CIM_HostedService 27

154 Table 15 – Operations: CIM_MetricDefForME..... 27

155 Table 16 – Operations: CIM_MetricForME 28

156 Table 17 – Operations: CIM_MetricInstance 28

157 Table 18 – Operations: CIM_ServiceAffectsElement 29

158 Table 19 – CIM Elements: Base Metrics Profile 47

159 Table 20 – Class: CIM_AggregationMetricDefinition 48

160 Table 21 – Class: CIM_AggregationMetricDefinition (Low Watermark) 48

161 Table 22 – Class: CIM_AggregationMetricDefinition (High Watermark) 48

162 Table 23 – Class: CIM_AggregationMetricValue 49

163 Table 24 – Class: CIM_BaseMetricDefinition 49

164 Table 25 – Class: CIM_BaseMetricDefinition – Instantaneous Metric..... 50

165 Table 26 – Class: CIM_BaseMetricDefinition – Interval Metric..... 50

166 Table 27 – Class: CIM_BaseMetricDefinition – Startup Interval Metric..... 50

167 Table 28 – Class: CIM_BaseMetricDefinition – Summation Metric 50

168 Table 29 – Class: CIM_BaseMetricDefinition – Current Data..... 51

169 Table 30 – Class: CIM_BaseMetricValue 51

170 Table 31 – Class: CIM_BaseMetricValue – Current Data 51

171 Table 32 – Class: CIM_BaseMetricValue – Interval Metrics..... 52

172 Table 33 – Class: CIM_BaseMetricValue – Startup Interval Metrics..... 52

173 Table 34 – Class: CIM_BaseMetricValue – Summation Metric..... 52

174 Table 35 – Class: CIM_BaseMetricValue – Long-Term Monitoring..... 52

175 Table 36 – Class: CIM_ConcreteDependency (Definition) 53

176 Table 37 – Class: CIM_ConcreteDependency (Value) 53

177 Table 38 – Class: CIM_ElementCapabilities..... 53

178 Table 39 – Class: CIM_HostedService 54

179 Table 40 – Class: CIM_MetricDefForME 54

180 Table 41 – Class: CIM_MetricForME 54

181 Table 42 – Class: CIM_MetricInstance 54

182 Table 43 – Class: CIM_MetricService..... 55

183	Table 44 – Class: CIM_MetricServiceCapabilities	55
184	Table 45 – Class: CIM_RegisteredProfile	56
185	Table 46 – Class: CIM_ServiceAffectsElement	56
186	Table B.1 – Simple Metric	59
187	Table B.2 – Summation Metric	61
188	Table B.3 – Aggregation Metric	63
189	Table B.4 – Aggregation Metric – Low Watermark	65
190	Table B.5 – Aggregation Metric – High Watermark	68
191		

192

Foreword

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

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

196 **Acknowledgments**

197 The authors wish to acknowledge the following people.

198 **Editors:**

- 199 • Oliver Benke – IBM
- 200 • Aaron Merkin – IBM
- 201 • Khachatur Papanyan – Dell

202 **Contributors:**

- 203 • Andreas Maier – IBM
- 204 • Karl Schopmeyer – The Open Group

205

Introduction

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

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

212

Base Metrics Profile

213 1 Scope

214 The *Base Metrics Profile* is a component profile that defines the minimum object model needed to provide
215 dynamic metrics associated to existing managed elements and related associations. This profile does not
216 document how to model metrics for capacity planning or accounting purposes. These topics are covered
217 by the *Capacity Metrics Profile* ([DSP1073](#)), which is a specialization of this profile.

218 2 Normative References

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

222 DMTF DSP0004, *CIM Infrastructure Specification 2.3*,
223 http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

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

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

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

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

232 3 Terms and Definitions

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

234 3.1

235 **can**

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

237 3.2

238 **cannot**

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

240 3.3

241 **conditional**

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

- 244 **3.4**
245 **mandatory**
246 indicates requirements to be followed strictly in order to conform to the document and from which no
247 deviation is permitted
- 248 **3.5**
249 **may**
250 indicates a course of action permissible within the limits of the document
- 251 **3.6**
252 **need not**
253 indicates a course of action permissible within the limits of the document
- 254 **3.7**
255 **optional**
256 indicates a course of action permissible within the limits of the document
- 257 **3.8**
258 **referencing profile**
259 indicates a profile that owns the definition of this class and can include a reference to this profile in its
260 "Referenced Profiles" table
- 261 **3.9**
262 **shall**
263 indicates requirements to be followed strictly in order to conform to the document and from which no
264 deviation is permitted
- 265 **3.10**
266 **shall not**
267 indicates requirements to be followed in order to conform to the document and from which no deviation is
268 permitted
- 269 **3.11**
270 **should**
271 indicates that among several possibilities, one is recommended as particularly suitable, without
272 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 273 **3.12**
274 **should not**
275 indicates that a certain possibility or course of action is deprecated but not prohibited
- 276 **3.13**
277 **unspecified**
278 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 279 **3.14**
280 **aggregation metric**
281 a type of metric that is derived by applying a formula or filter to a set of base metric values
- 282 **3.15**
283 **base metric**
284 a metric provided directly without a dependency on other metric values

- 285 **3.16**
286 **measured resource**
287 a managed object being measured, which is the resource to which base metric value instances are
288 associated
- 289 **3.17**
290 **sampling interval**
291 a value that determines how often new metric values are retrieved, if metrics are retrieved periodically
- 292 **3.18**
293 **current data**
294 the most current data available for a given metric. Online monitoring (3.19) and snapshot monitoring
295 (3.20) are types of current data access.
- 296 **3.19**
297 **online monitoring**
298 the process in which metric values (typically interval metrics) are gathered asynchronously to a request
299 from the instrumentation or reporting layer
- 300 **3.20**
301 **snapshot monitoring**
302 the process in which metric values are gathered synchronously with a request from the instrumentation or
303 reporting layer
- 304 **3.21**
305 **long-term monitoring**
306 the process in which metric values are captured during an interval
- 307 **3.22**
308 **event-based monitoring**
309 the process in which threshold values for metrics are used to trigger asynchronous notification
- 310 **3.23**
311 **instantaneous metrics**
312 metrics that apply to a particular point in time. An example of an instantaneous metric is the amount of
313 memory currently allocated to a virtual server.
- 314 **3.24**
315 **interval metrics**
316 metrics that apply to a time interval. An example of an interval metric is the average CPU utilization of a
317 server over the past hour.
- 318 **3.25**
319 **summation metrics**
320 a type of counter metric that reflects the accumulation of a value
- 321 **3.26**
322 **watermark metrics**
323 a type of aggregation metric used to capture the minimum or maximum value recorded for a monitored
324 value

325 4 Symbols and Abbreviated Terms

326 4.1

327 CPU

328 central processing unit

329 4.2

330 IEPS

331 instructions executed per second

332 4.3

333 UTC

334 Universal Time Coordinated

335 4.4

336 UUID

337 Universally Unique Identifier

338 5 Synopsis

339 **Profile Name:** Base Metrics

340 **Version:** 1.0.1

341 **Organization:** DMTF

342 **CIM Schema Version:** 2.23

343 **Central Class:** CIM_MetricService

344 **Scoping Class:** CIM_System

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

346

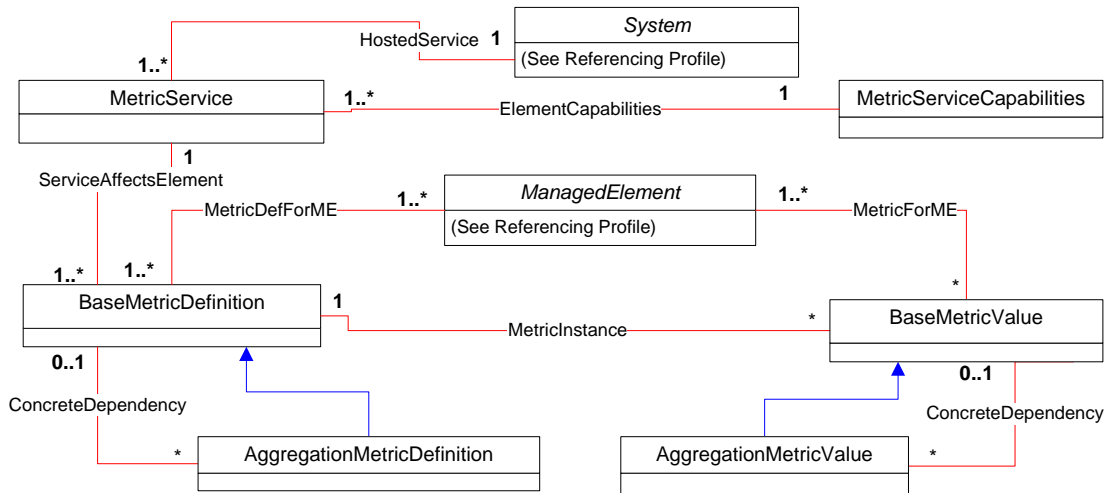
Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

347 6 Description (Informative)

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

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



351

352

Figure 1 – Base Metrics Profile: Class Diagram

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

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

- 361 • Metric access type
- 362 • Time scope of the metric
- 363 • Formulation of the metric value
- 364 • Metric context

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

369 6.1 Metric Access Types

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

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

375 6.1.1 Current Data

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

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

381 **6.1.1.1 Online Monitoring**

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

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

389 **6.1.1.2 Snapshot Monitoring**

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

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

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

398 **6.1.2 Long-Term Monitoring**

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

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

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

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

406 **6.1.3 Event-Based Monitoring**

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

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

412 **6.2 Metric Time Scope**

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

417 **6.2.1 Instantaneous Metrics**

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

421 **6.2.2 Interval Metrics**

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

426 **6.2.3 Startup Interval Metrics**

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

431 **6.3 Metric Value Formulation**

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

435 **6.3.1 Simple Metrics**

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

439 **6.3.2 Summation Metrics**

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

443 More information on using summation metrics is specified in [DSP1073](#).

444 **6.3.3 Aggregation Metrics**

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

455 **6.3.3.1 Watermark Metrics**

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

459 **6.4 Metric Context**

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

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

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

478 **7 Implementation**

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

481 **7.1 Common Requirements**

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

485 **7.1.1 Service and Capabilities**

486 At least one instance of CIM_MetricService shall exist. Each instance of CIM_MetricService shall be
487 associated with exactly one instance of CIM_System through the CIM_HostedService association. Each
488 instance of CIM_MetricService shall be associated with exactly one instance of
489 CIM_MetricServiceCapabilities through the CIM_ElementCapabilities association. Each instance of
490 CIM_BaseMetricDefinition shall be associated with exactly one instance of CIM_MetricService through
491 the CIM_ServiceAffectsElement association.

492 **7.1.2 Relating a Metric Definition and Metric Value**

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

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

497 7.1.3 Identifying a Metric Definition

498 Incorporating profiles may specify metric definitions for metrics that are applicable to the management
499 domain of the incorporating profile.

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

502 "DMTF:<unique identifier> "

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

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

512 7.1.4 Identifying Metric Context

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

514 7.1.4.1 General Requirements

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

520 7.1.4.2 Breakdown Dimensions (Optional)

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

526 The CIM_BaseMetricDefinition.BreakdownDimensions property shall not be NULL.

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

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

532 <schemaName>"_ "<simpleClassName>

533 as specified in [DSP0004](#).

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

537 "DMTF" <unique identifier>

538

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

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

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

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

555 7.1.5 Gathering Type

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

560 7.2 Modeling Metric Access Types

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

564 7.2.1 Modeling Current Data Access Type (Optional)

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

568 7.3 Modeling Metric Time Scope

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

572 7.3.1 Modeling Instantaneous Metrics (Optional)

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

576 7.3.2 Modeling Interval Metrics (Optional)

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

579 **7.3.3 Modeling Interval Metrics (Optional)**

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

582 **7.4 Modeling Metric Value Formulation**

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

584 **7.4.1 Modeling Summation Metrics (Optional)**

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

587 **7.4.2 Modeling Aggregation Metrics (Optional)**

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

592 **7.4.2.1 Modeling Low Watermark Metrics (Optional)**

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

595 **7.4.2.2 Modeling High Watermark Metrics (Optional)**

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

598 **7.5 Relationship between Aggregation and Base Metrics**

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

608 **7.6 Constraints on Metric Values for Controllable Metrics**

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

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

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

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

627 **8 Methods**

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

632 **8.1 CIM_MetricService.ShowMetrics()**

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

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

638 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

640

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

641 **8.1.1 CIM_MetricService.ShowMetrics() Conditional Support**

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

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

648 **8.2 CIM_MetricService.ShowMetricsByClass()**

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

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

654 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

656

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

657

8.2.1 CIM_MetricService.ShowMetricsByClass() Conditional Support

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

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

664

8.3 CIM_MetricService.ControlMetrics()

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

- 666 • a metric for all instances of CIM_ManagedElement
- 667 • all metrics for a single CIM_ManagedElement instance
- 668 • a single metric for a single CIM_ManagedElement instance

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

672 No standard messages are defined for this method.

673

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

674

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

675 **8.3.1 CIM_MetricService.ControlMetrics() Conditional Support**

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

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

682 **8.3.2 Parameter Validation**

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

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

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

- 695 • The instance of CIM_ManagedElement identified by the Subject parameter is identified by a
 696 value of the ControllableManagedElements property of the associated instance of
 697 CIM_MetricServiceCapabilities, where the corresponding array index of the
 698 ManagedElementControlTypes property of the CIM_MetricServiceCapabilities instance has a
 699 value of 2 (Discrete), and the instance of CIM_BaseMetricDefinition identified by the Definition
 700 parameter is identified by a value of the ControllableMetrics property of the associated instance
 701 of CIM_MetricServiceCapabilities, where the corresponding array index of the
 702 MetricsControlTypes property of the CIM_MetricServiceCapabilities instance has a value of 2
 703 (Discrete).

- 704 • The instance of CIM_BaseMetricDefinition identified by the Definition parameter is identified by
 705 a value of the ControllableMetrics property of the associated instance of
 706 CIM_MetricServiceCapabilities, where the corresponding array index of the
 707 MetricsControlTypes property of the CIM_MetricServiceCapabilities instance has a value of 2
 708 (Discrete) and no instances of CIM_ManagedElement that are associated with the
 709 CIM_BaseMetricDefinition through the CIM_MetricDefForME are identified by a value of the
 710 ControllableManagedElements property of the associated instance of
 711 CIM_MetricServiceCapabilities.

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

715 8.4 CIM_MetricService.ControlMetricsByClass()

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

- 717 • a metric for all instances of a specific CIM class
- 718 • all metrics for all instances of a specific CIM class
- 719 • a single metric for a single CIM_ManagedElement

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

723 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

725 **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

726 8.4.1 CIM_MetricService.ControlMetricsByClass() Conditional Support

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

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

733 8.4.2 Parameter Validation

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

739 **8.5 CIM_MetricService.GetMetricValues()**

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

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

744 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

746 **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

747 **8.5.1 CIM_MetricService.GetMetricValues() Conditional Support**

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

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

754 **8.6 Profile Conventions for Operations**

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

757 The default list of operations is as follows:

- 758 • GetInstance
- 759 • Associators
- 760 • AssociatorNames
- 761 • References

- 762 • ReferenceNames
- 763 • EnumerateInstances
- 764 • EnumerateInstanceNames

765 8.7 CIM_AggregationMetricDefinition

766 All operations in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

768 8.8 CIM_AggregationMetricValue

769 All operations in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

771 8.9 CIM_BaseMetricDefinition

772 All operations in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

774 8.10 CIM_BaseMetricValue

775 All operations in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

777 8.11 CIM_ConcreteDependency

778 Table 12 lists implementation requirements for operations. If implemented, these operations shall be
 779 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 12, all operations
 780 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

782 **Table 12 – Operations: CIM_ConcreteDependency**

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

783 8.12 CIM_ElementCapabilities

784 Table 13 lists implementation requirements for operations. If implemented, these operations shall be
 785 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 13, all operations
 786 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

788

Table 13 – Operations: CIM_ElementCapabilities

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

789 **8.13 CIM_HostedService**

790 Table 14 lists implementation requirements for operations. If implemented, these operations shall be
 791 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 14, all operations
 792 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

794

Table 14 – Operations: CIM_HostedService

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

795 **8.14 CIM_MetricDefForME**

796 Table 15 lists implementation requirements for operations. If implemented, these operations shall be
 797 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 15, all operations
 798 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

800

Table 15 – Operations: CIM_MetricDefForME

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

801 **8.15 CIM_MetricForME**

802 Table 16 lists implementation requirements for operations. If implemented, these operations shall be
 803 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 16, all operations
 804 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

806 **Table 16 – Operations: CIM_MetricForME**

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

807 **8.16 CIM_MetricInstance**

808 Table 17 lists implementation requirements for operations. If implemented, these operations shall be
 809 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 17, all operations
 810 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

812 **Table 17 – Operations: CIM_MetricInstance**

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

813 **8.17 CIM_MetricService**

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

815 **8.18 CIM_MetricServiceCapabilities**

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

817 8.19 CIM_ServiceAffectsElement

818 Table 18 lists implementation requirements for operations. If implemented, these operations shall be
 819 implemented as defined in [DSP0200](#). In addition, and unless otherwise stated in Table 18, all operations
 820 in the default list in 8.6 shall be implemented as defined in [DSP0200](#).

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

822 **Table 18 – Operations: CIM_ServiceAffectsElement**

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

823 9 Use Cases (Informative)

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

825 9.1 Instructions Executed per Second

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

829 9.1.1 Interval Metrics

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

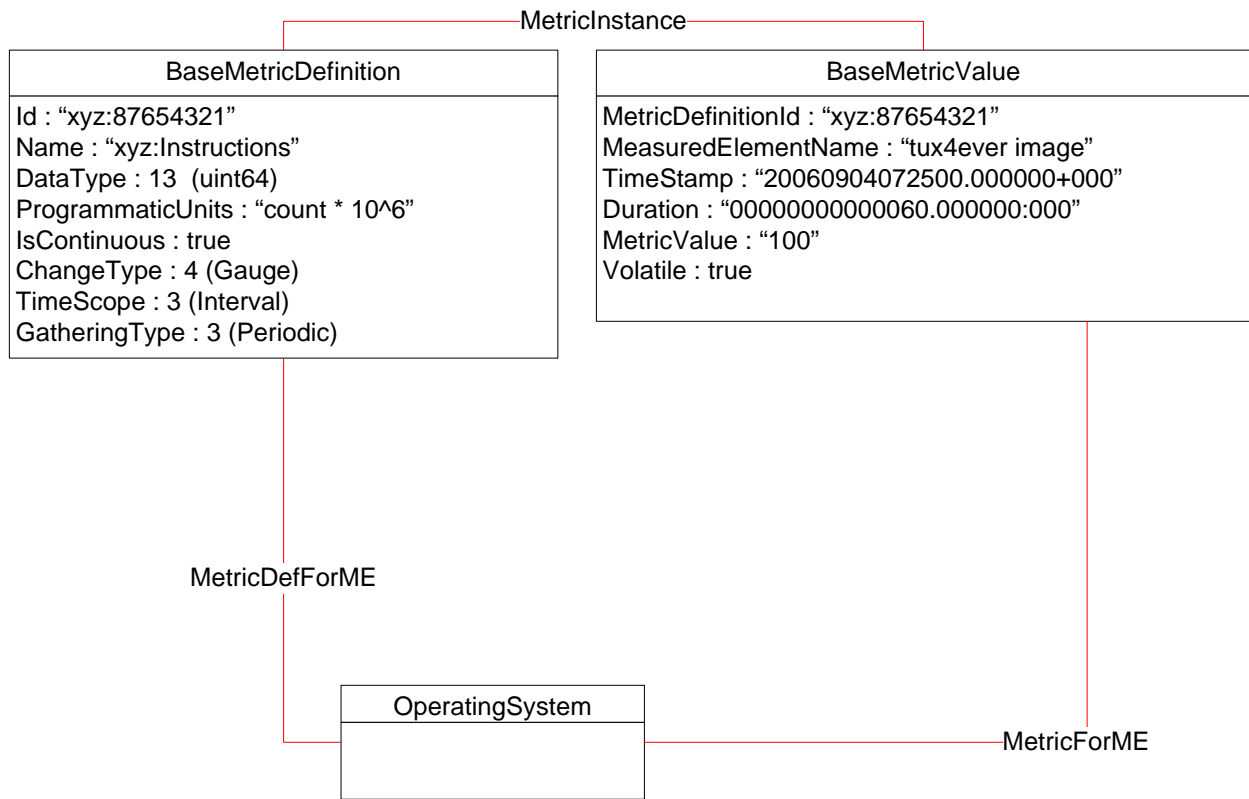
838 The CIM_BaseMetricDefinition.Id property contains a UUID that is chosen by the metrics provider.

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

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

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

847



848

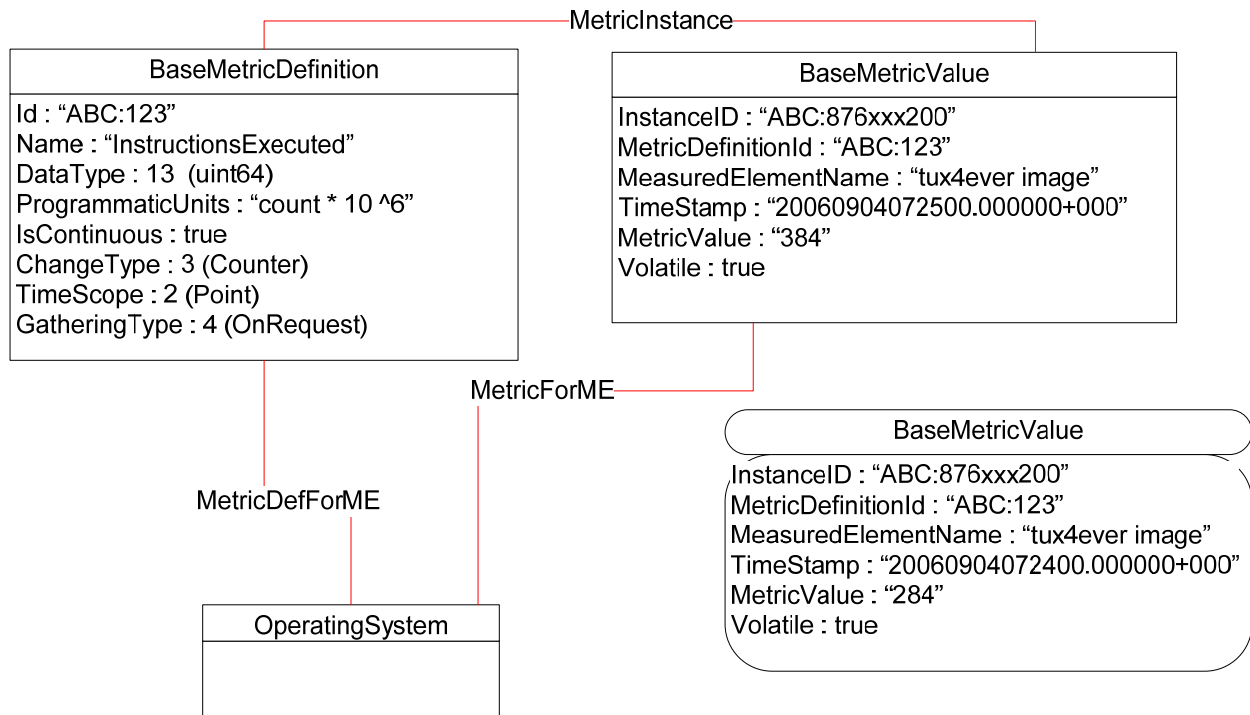
849

Figure 2 – Interval Metrics

850 **9.1.2 Instantaneous Counter**

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

858



859

860

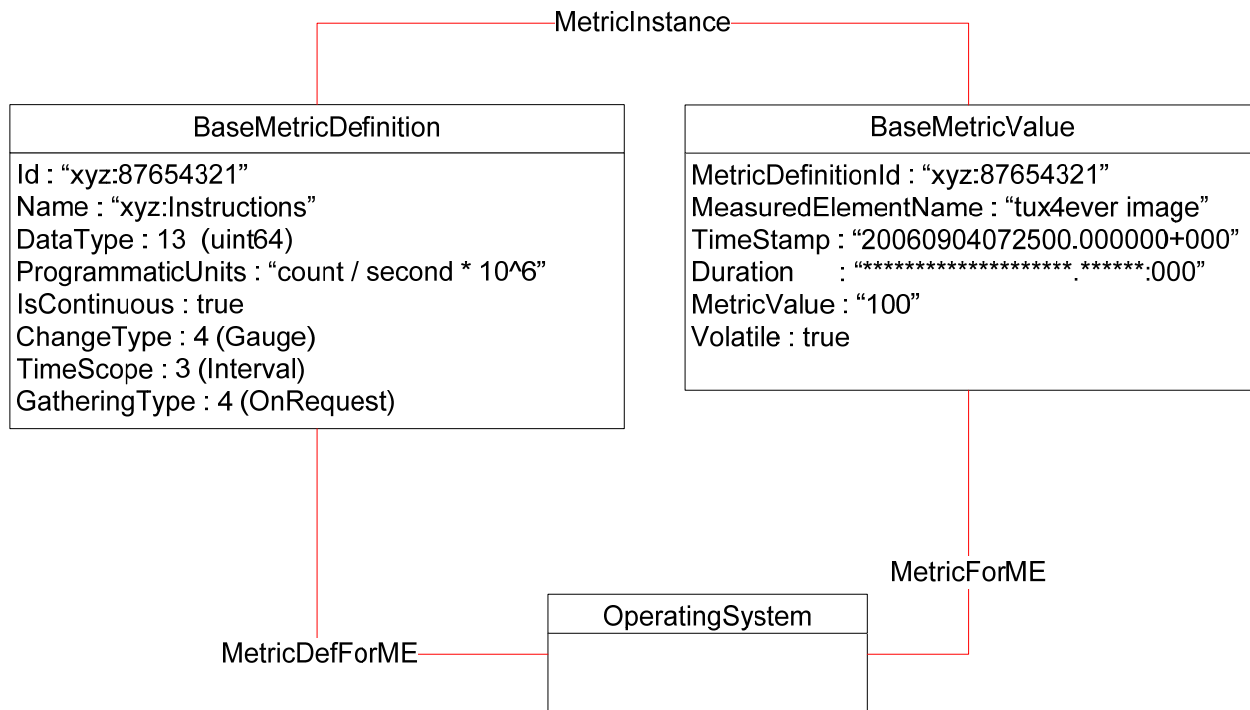
Figure 3 – Instantaneous Counter

861 **9.1.3 Instantaneous Gauge**

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

866 The ProgrammaticUnits property indicates that the metric reports millions of instructions per second. The
 867 CIM_BaseMetricDefinition.TimeScope property indicates that the metric is an interval metric. The
 868 CIM_BaseMetricValue.Duration property indicates that there is no precision to the reported interval
 869 duration. The current values of the properties of the CIM_BaseMetricValue instance indicate that, as of
 870 07:25:00 A.M. at 9/4/2006, an average of 100 million instructions were executed per second.

871



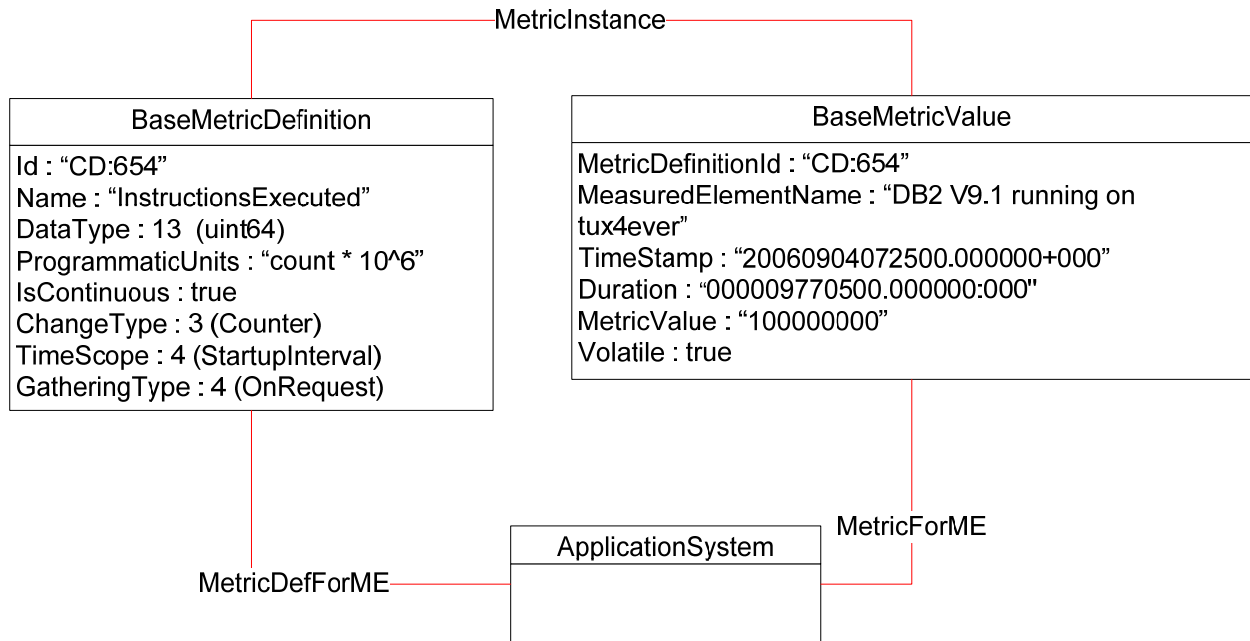
872

873

Figure 4 – Instantaneous Gauge

874 **9.2 Object Diagram for Startup Interval Time Scope**

875 A value of 4 (StartupInterval) for the TimeScope property indicates that the metric applies to an interval
 876 that began at the startup of the measured resource. The example in Figure 5 shows that at 07:25:00 A.M.
 877 on 09/04/2006, the associated application system "DB2 V9.1 on tux4ever" was running for a duration of
 878 977 days and 5 hours, consuming 100 million resources. The associated metric is "InstructionsExecuted",
 879 with a unit of "Million Count" of instructions.



880

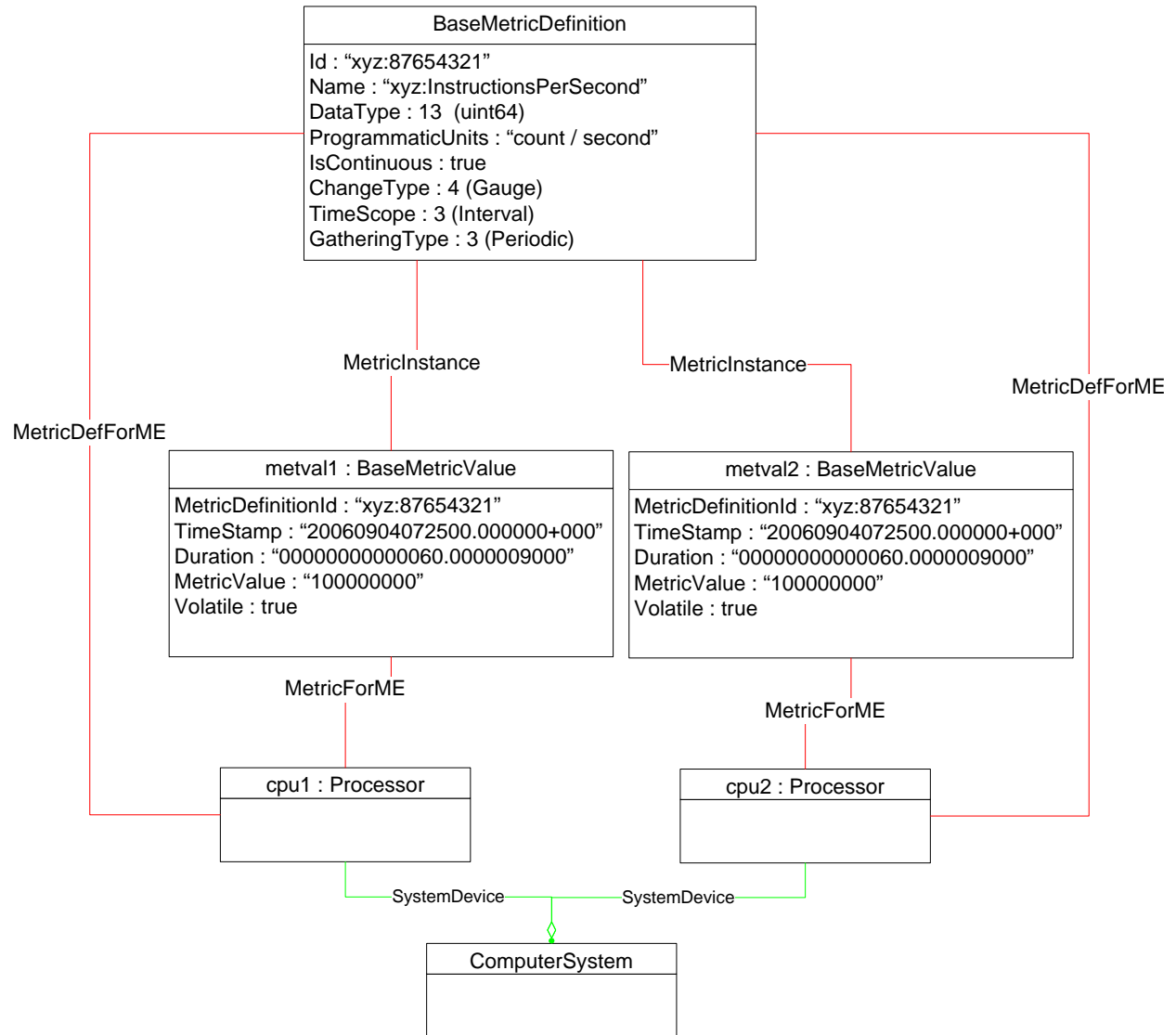
881

Figure 5 – Usage Example for Startup Interval Time Scope

882 **9.3 Metric Definition for Multiple Instances of CIM_ManagedElement**

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

886

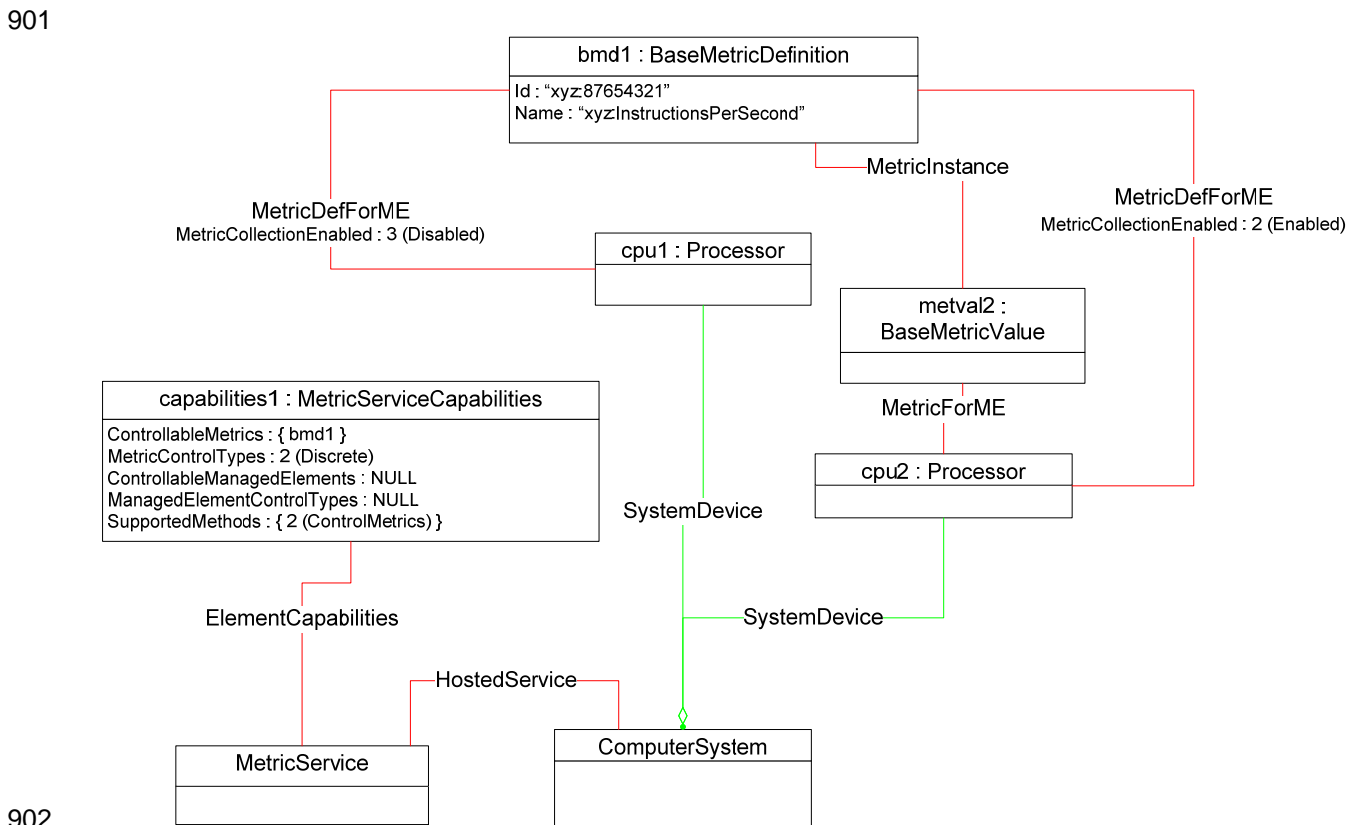


887

888 **Figure 6 – Common Metric Definition for Multiple Instances of CIM_ManagedElement**

889 **9.4 Controllable Metrics**

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

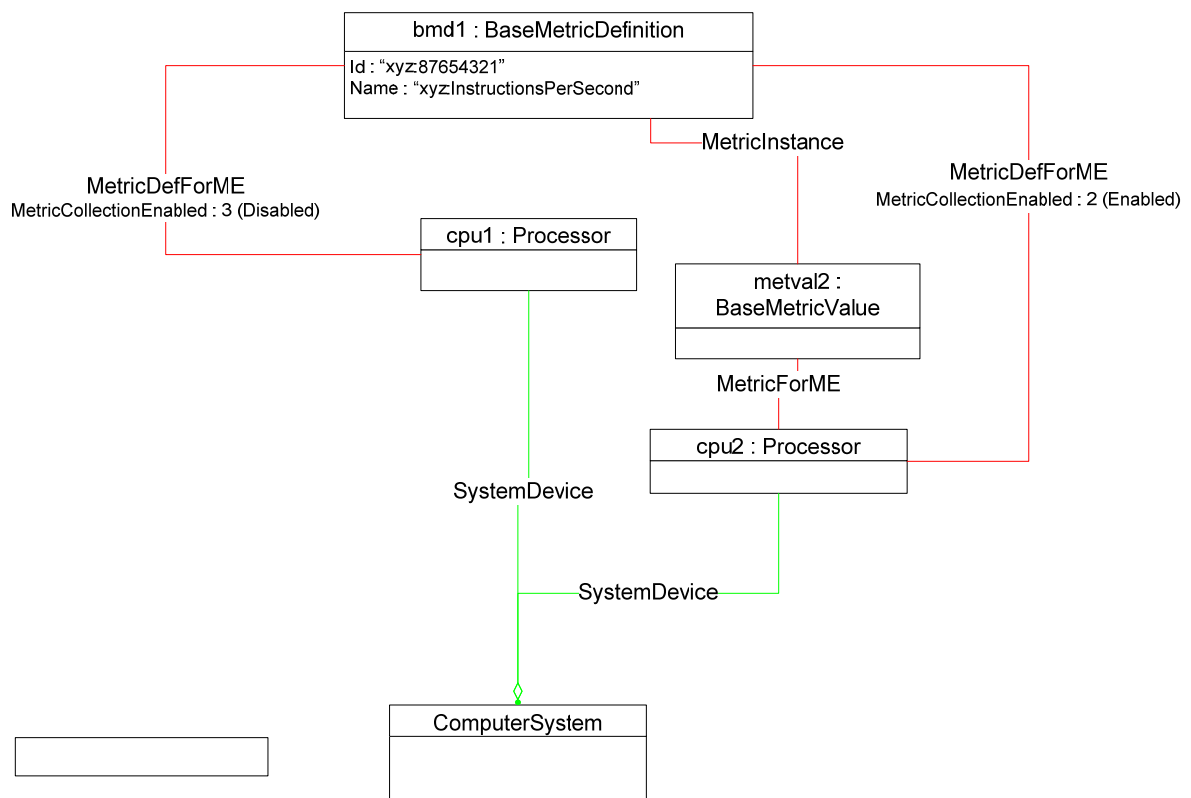


903 **Figure 7 – Advertising Support for Discrete Controllable Metrics**

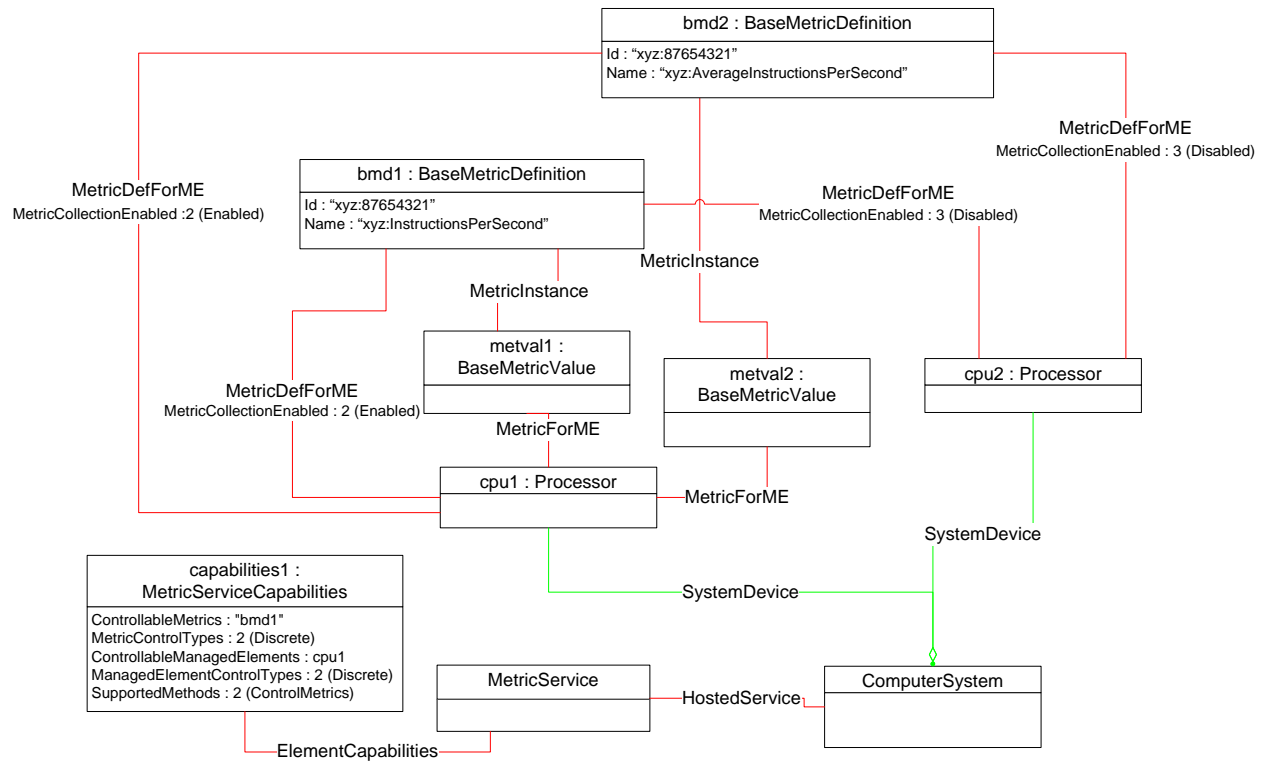
904 Figure 8 shows an object diagram for an implementation in which a single metric, represented with the
 905 CIM_BaseMetricDefinition instance bmd1, is available for two processors, represented by the
 906 CIM_Processor instances cpu1 and cpu2.

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

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



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



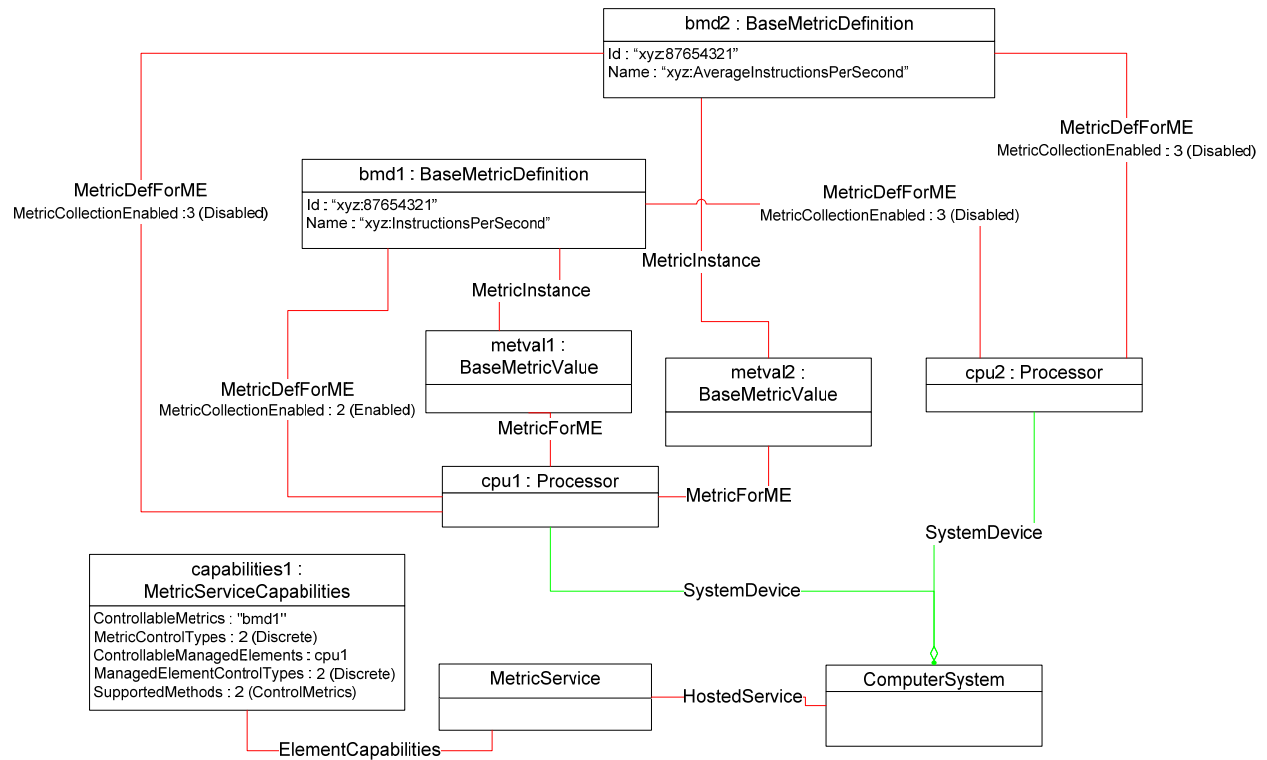
924

925

Figure 9 – Discrete Controllable Metrics (After Enable)

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

935

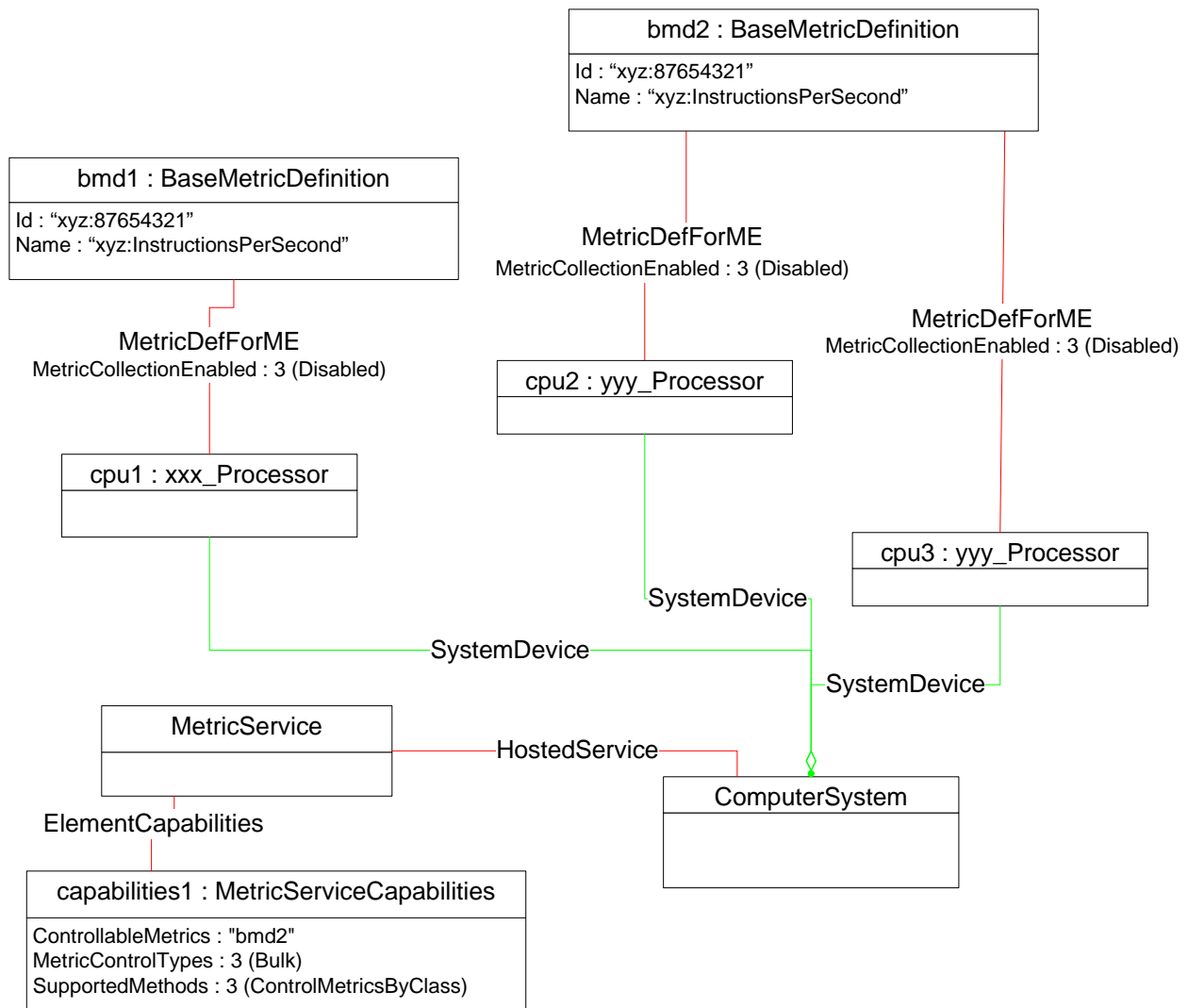


936

937

Figure 10 – Bulk Controllable Metrics by Definition

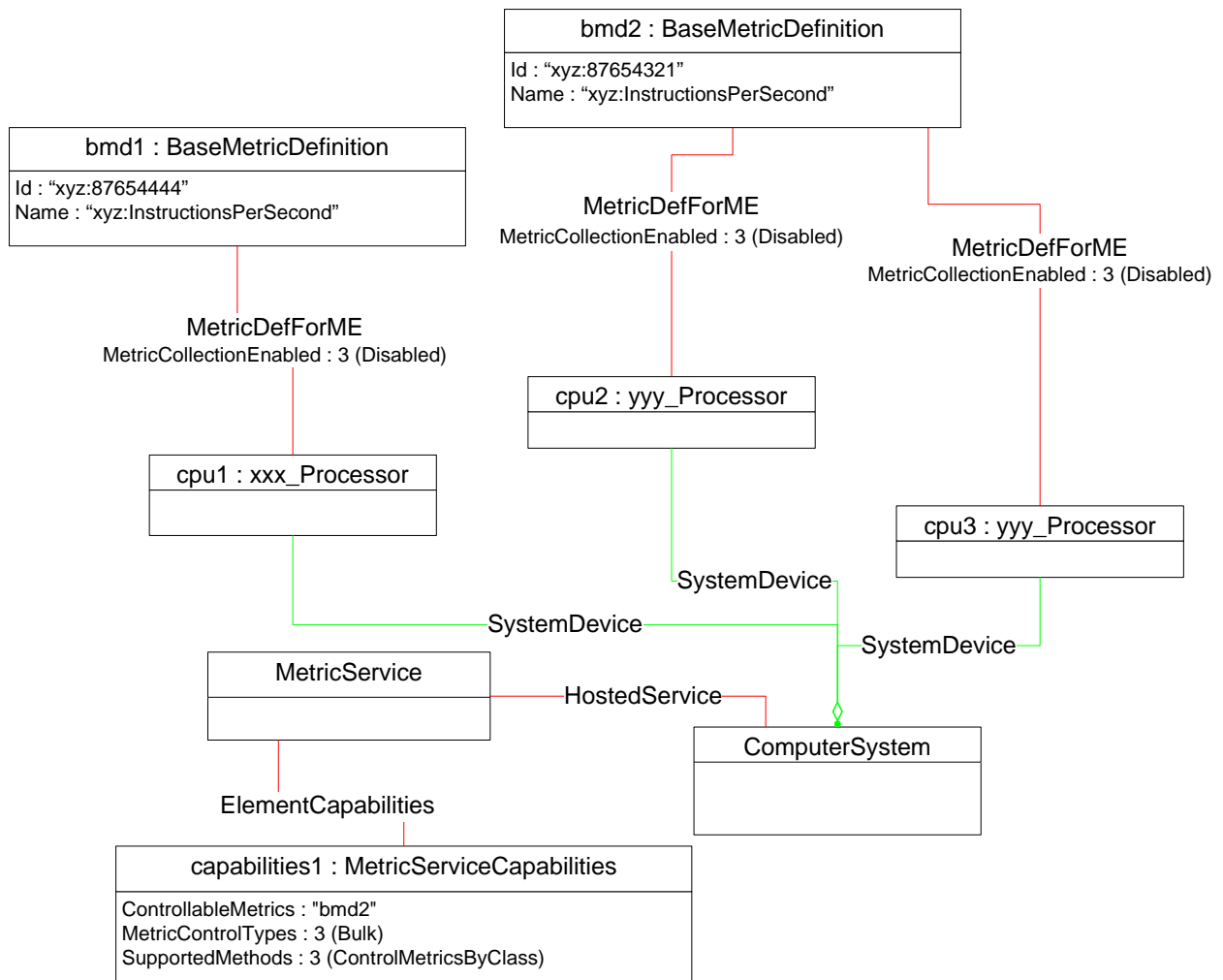
938 Figure 11 shows an object diagram for an implementation in which two metrics, represented by the
 939 CIM_BaseMetricDefinition instances bmd1 and bmd2, are available for three processors, represented by
 940 CIM_Processor instances cpu1, cpu2, and cpu3. The collection of all metric values for bmd2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



948 **Figure 11 – Bulk Controllable Metrics by Managed Element**

949 Figure 12 shows an object diagram for an implementation in which a single metric is available for three
 950 processors, represented by instances of subclasses of CIM_Processor cpu1, cpu2, and cpu3. Two
 951 instances of CIM_BaseMetricDefinition (bmd1 and bmd2) define the same standard metric
 952 "xyz:InstructionsPerSecond". Multiple instances of the CIM_BaseMetricDefinition class are required in
 953 order to represent the separate control points for collection of the metric values. The collection of bmd2 is
 954 controlled for all instances of the yyy_Processor class as a bulk operation. Control of the collection of the
 955 metric value defined by bmd1 for cpu1 is not supported. The CIM_ServiceAffectsElement associations
 956 between the CIM_MetricService instance and the CIM_BaseMetricDefinition instances have been elided.

957



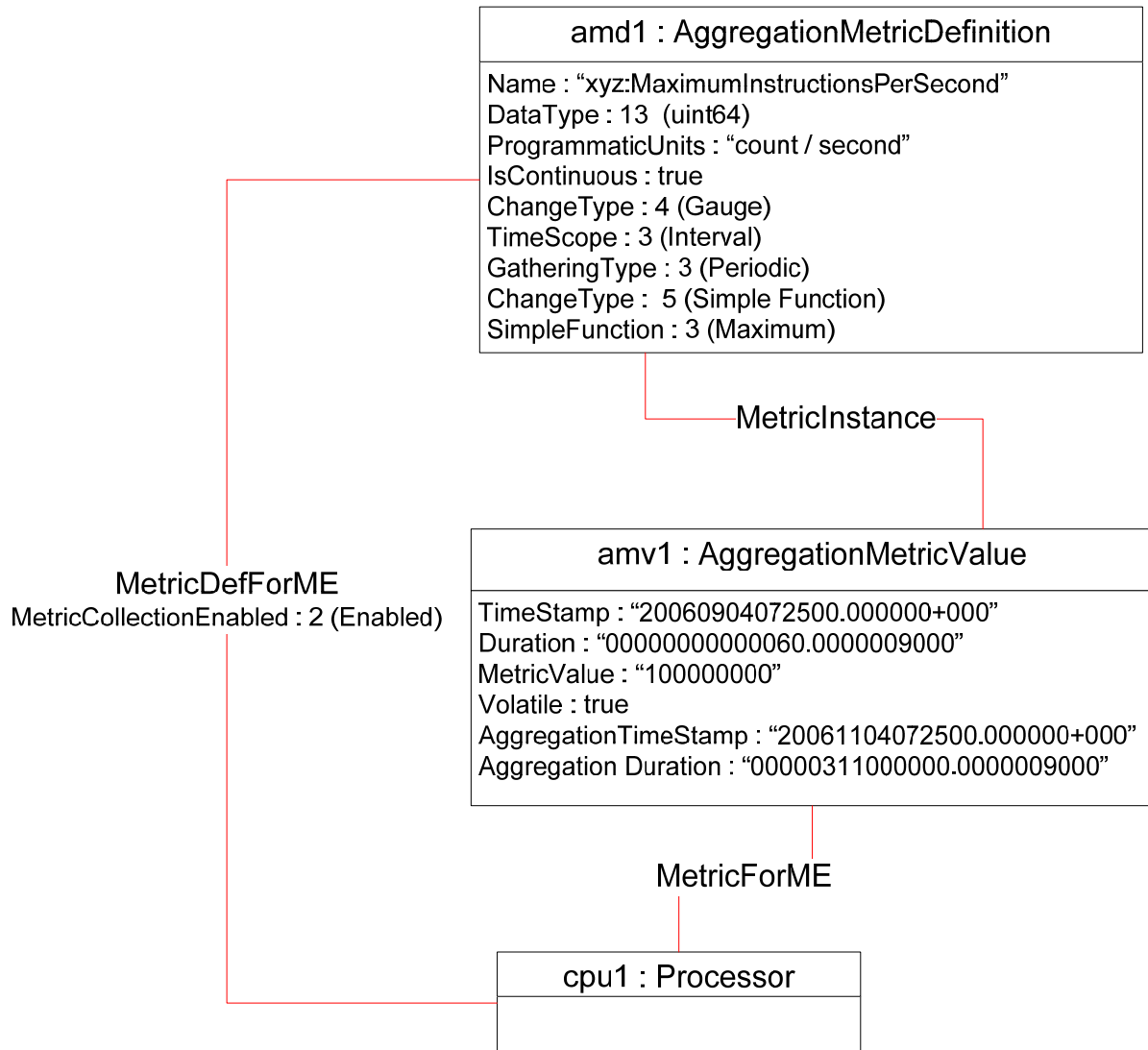
958

959

Figure 12 – Bulk Controllable Metrics by Class

960 **9.5 Aggregation Metrics**

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

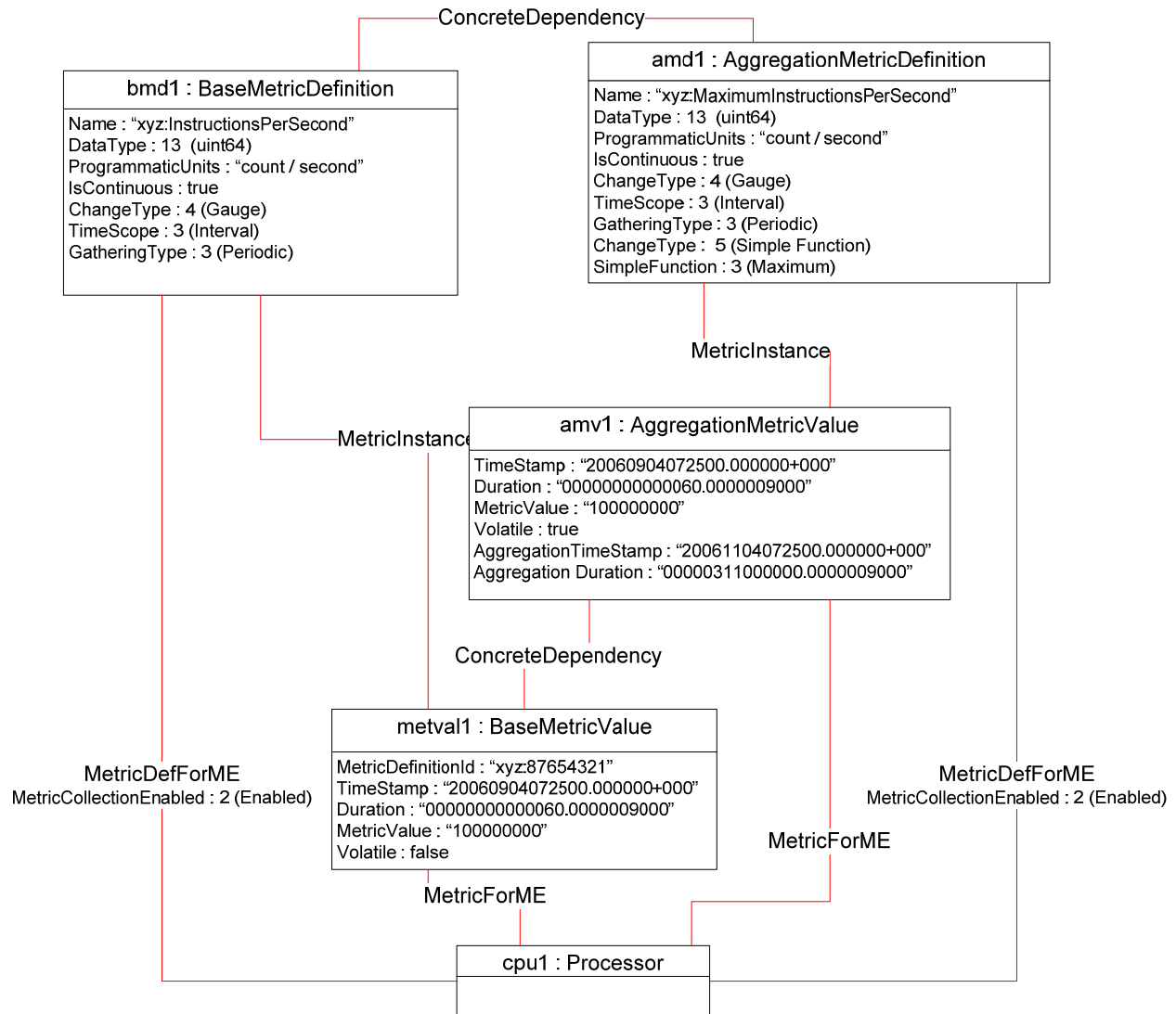


964

965

Figure 13 – Aggregation Metric without Base

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



971

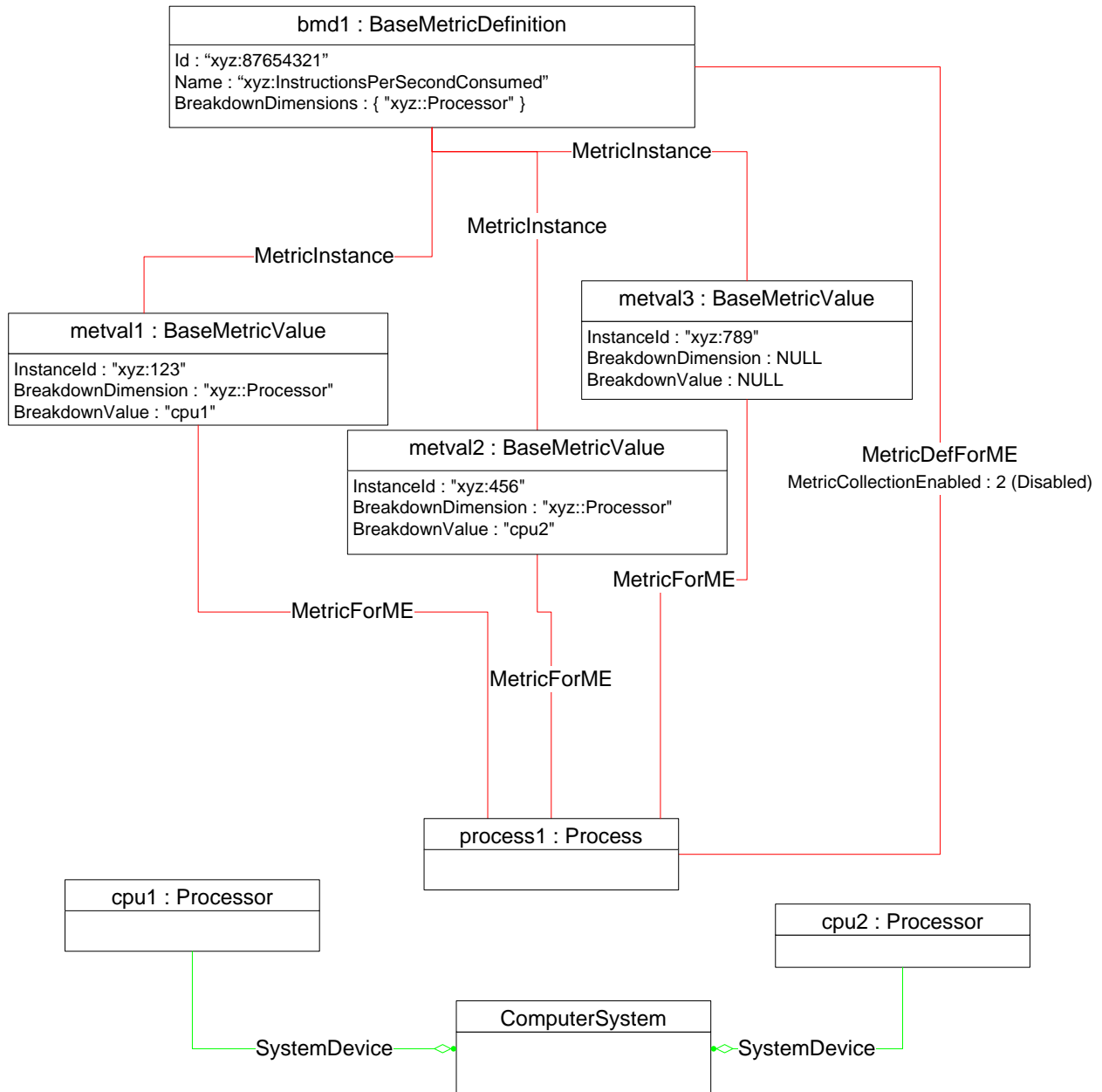
972

Figure 14 – Aggregation Metric with Base

973 **9.6 Metric Context**

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

979



980

981

Figure 15 – Metric Context

982 **9.7 Find All Metric Definitions for a Managed Element**

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

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

987 **9.8 Find the Metric Value for a Managed Element**

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

- 991 1) Find the instance of CIM_MetricDefForME that associates the CIM_BaseMetricDefinition with
992 the CIM_ManagedElement instance.
- 993 2) If the value of the MetricCollectionEnabled property of the instance of CIM_MetricDefForME
994 association found in the previous step is 2 (Enabled):
- 995 a) Find all instances of CIM_BaseMetricValue associated with the CIM_BaseMetricDefinition
996 through the CIM_MetricInstance association.
- 997 b) Find all instances of CIM_BaseMetricValue associated with the CIM_ManagedElement
998 instance through CIM_MetricForME.
- 999 c) Find the instance of CIM_BaseMetricValue that is the intersection of the two result sets by
1000 finding matching InstanceID property values.
- 1001 3) Otherwise, metric collection is disabled and a current data metric value is not available.

1002 **9.9 Find a Standard Metric for a Managed Element**

1003 Given a string value corresponding to the unique identifier of a standard metric, a client can find the
1004 standard metric value for an instance of CIM_ManagedElement as follows:

- 1005 1) Use the steps in 9.6 to find all metric definitions available for the instance of
1006 CIM_ManagedElement.
- 1007 2) For each instance of CIM_BaseMetricDefinition returned, determine if the Name property
1008 matches the string identifier. If there is a match, use the steps in 9.8 to find the metric value.
- 1009 3) If a matching Name property is not found, the standard metric is not supported for the instance
1010 of CIM_ManagedElement.

1011 **9.10 Retrieve a Metric Value**

1012 A client can retrieve a metric value as follows:

- 1013 1) Using the steps in 9.9, find the instance of CIM_BaseMetricValue that reports the metric.
- 1014 2) Invoke GetInstance to query the current values of properties of the CIM_BaseMetricValue
1015 instance.

1016 **9.11 Find All Metrics Available for a Managed Element within an Enumeration**
 1017 **Scope**

1018 Given an instance of CIM_ManagedElement, a client can find all of the metrics available for an instance
 1019 of CIM_ManagedElement as follows:

- 1) Enumerate all instances of CIM_MetricService within the enumeration scope.
 - 1020 a) For each instance of CIM_MetricService, find the instance of
 1021 CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities.
 - 1022 b) Query the value of the CIM_MetricServiceCapabilities.SupportedMetrics property.
 - 1023 c) If the array contains the value 4 (Show Metrics), invoke the
 1024 CIM_MetricService.ShowMetrics() method providing the reference to the
 1025 CIM_ManagedElement.
 - 1026 d) The list of references to CIM_BaseMetricDefinition returned as the value of the Definitions
 1027 parameter identifies instances of CIM_BaseMetricDefinition that are available for the
 1028 CIM_ManagedElement instance.
- 1029 2) Starting with a reference to the instance of CIM_ManagedElement, find all instances of
 1030 CIM_MetricDefinition that are associated through CIM_MetricDefForME.
- 1031 3) Union the results of d) and 2).

1032 **9.12 Find All Metrics Available within an Enumeration Scope for All Instances of a**
 1033 **CIM Class**

1034 Given a CIM class name, a client can find all of the metrics available within an enumeration scope for all
 1035 instances of the class as follows:

- 1036 1) Enumerate all instances of CIM_MetricService within the enumeration scope.
 - 1037 a) For each instance of CIM_MetricService, find the instance of
 1038 CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities.
 - 1039 b) Query the value of the CIM_MetricServiceCapabilities.SupportedMetrics property.
 - 1040 c) If the array contains the value 5 (ShowMetricsByClass), invoke the
 1041 CIM_MetricService.ShowMetrics() method providing the reference to the
 1042 CIM_ManagedElement.
 - 1043 d) The list of references to CIM_BaseMetricDefinition returned as the value of the Definitions
 1044 parameter identifies instances of CIM_BaseMetricDefinition that are available for the
 1045 CIM_ManagedElement instance.
- 1046 2) Enumerate all instances of the CIM class.
 - 1047 a) For each instance of the CIM class, find all instances of CIM_BaseMetricDefinition that are
 1048 associated through CIM_MetricDefForME.
- 1049 3) Form a set of instances of CIM_BaseMetricDefinition from the intersection of the instances
 1050 returned by a).
- 1051 4) Union the results of d) and 3).

1052 **9.13 Determine whether a Metric Can Be Discretely Controlled for a Specific** 1053 **Managed Element**

1054 Given an instance of CIM_BaseMetricDefinition associated to an instance of CIM_ManagedElement
1055 through CIM_MetricDefForME, a client can determine whether the metric can be controlled for the
1056 managed element as follows:

- 1057 1) Starting with the instance of CIM_BaseMetricDefinition, find the instance of CIM_MetricService
1058 associated through CIM_ServiceAffectsElement.
- 1059 2) Find the instance of CIM_MetricServiceCapabilities associated through
1060 CIM_ElementCapabilities with the instance of CIM_MetricService found in 1).
- 1061 3) If the following conditions are met, the metric can be discretely controlled for the managed
1062 element:
 - 1063 a) The CIM_MetricServiceCapabilities.ControllableMetrics property contains a reference to
1064 the CIM_BaseMetricDefinition.
 - 1065 b) The CIM_MetricServiceCapabilities.MetricControlTypes property contains the value 2
1066 (Discrete) or 4 (Both) at the same array index as the reference in the previous step.
 - 1067 c) The CIM_MetricServiceCapabilities.ControllableManagedElements property contains a
1068 reference to the CIM_ManagedElement.
 - 1069 d) The CIM_MetricServiceCapabilities.ManagedElementControlTypes property contains the
1070 value 2 (Discrete) or 4 (Both) at the same array index as the reference in the previous step.
- 1071 4) Otherwise, the metric can not be discretely controlled for the managed element.

1072 **9.14 Enable a Specific Metric for a Specific Managed Element**

1073 Given an instance of CIM_BaseMetricDefinition that is associated to an instance of
1074 CIM_ManagedElement through the CIM_MetricDefForME association, a client can enable a specific
1075 metric for the managed element as follows:

- 1076 1) Use the steps in 9.13 to determine whether the metric can be controlled.
- 1077 2) Invoke the CIM_MetricService.ControlMetrics() method and specify the reference to the
1078 CIM_ManagedElement as the value of the Subject parameter, the reference to the
1079 CIM_BaseMetricDefinition as the value of the Definition parameter, and TRUE as the value of
1080 the MetricCollectionEnabled parameter.

1081 **9.15 Find All Managed Elements within an Enumeration Scope for which a Metric** 1082 **Is Currently Being Collected**

1083 Given an instance of CIM_BaseMetricDefinition, a client can find all instances of CIM_ManagedElement
1084 for which the metric is available as follows:

- 1085 1) Find the instance of CIM_MetricService associated with the CIM_BaseMetricDefinition through
1086 the CIM_ServiceAffectsElement association.
- 1087 2) Find the instance of CIM_MetricServiceCapabilities associated with the CIM_MetricService
1088 found in the previous step.
- 1089 3) Query the value of the SupportedMethods property of the instance of
1090 CIM_MetricServiceCapabilities found in the previous step to determine if it contains a value of 4
1091 (ShowMetrics):
 - 1092 a) If the SupportedMethods property contains a value of 4 (ShowMetrics):
 - 1093 – Invoke the CIM_MetricService.ShowMetrics() method, specifying the reference to the
1094 CIM_BaseMetricDefinition as the value of the Definition parameter.

- 1095 – Upon successful completion of the method, the ManagedElements parameter
- 1096 contains a list of references to CIM_ManagedElement instances for which the metric
- 1097 defined by the CIM_BaseMetricDefinition is available. The MetricCollectionEnabled
- 1098 property indicates whether the metric is currently being collected for the
- 1099 CIM_ManagedElement instance.

- 1100 4) If the SupportedMethods property does not contain the value 4, find all instances of
- 1101 CIM_MetricDefForME that reference the CIM_BaseMetricDefinition instance. For each instance
- 1102 of CIM_MetricDefForME, the Antecedent property identifies a CIM_ManagedElement for which
- 1103 the metric may be collected and the MetricCollectionEnabled property indicates whether the
- 1104 metric is currently being collected.

1105 **10 CIM Elements**

1106 Table 19 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 1107 implemented as described in Table 19. Sections 7 ("Implementation") and 8 ("Methods") may impose
 1108 additional requirements on these elements.

1109 **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		

1110 **10.1 CIM_AggregationMetricDefinition**

1111 CIM_AggregationMetricDefinition defines a metric that can be captured. Table 20 defines the
 1112 requirements for instances of CIM_AggregationMetricDefinition. This class is used as a basis for 10.2 and
 1113 for 10.3.

1114

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

1115 10.2 CIM_AggregationMetricDefinition (Low Watermark)

1116 Table 21 defines the requirements for instances of CIM_AggregationMetricDefinition used to define a low
1117 watermark metric. These constraints are in addition to those specified in 10.1.

1118

Table 21 – Class: CIM_AggregationMetricDefinition (Low Watermark)

Properties	Requirement	Notes
SimpleFunction	Mandatory	Matches 2 (Minimum)

1119 10.3 CIM_AggregationMetricDefinition (High Watermark)

1120 Table 22 defines the requirements for instances of CIM_AggregationMetricDefinition used to define high
1121 watermark metrics. These constraints are in addition to those specified in 10.1.

1122

Table 22 – Class: CIM_AggregationMetricDefinition (High Watermark)

Properties	Requirement	Notes
SimpleFunction	Mandatory	Matches 3 (Maximum)

1123 **10.4 CIM_AggregationMetricValue**

1124 CIM_AggregationMetricValue conveys the actual recorded data of a metric that has been maintained.

1125 Table 23 describes the requirements for instances of CIM_AggregationMetricValue.

1126 **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

1127 **10.5 CIM_BaseMetricDefinition**

1128 CIM_BaseMetricDefinition defines a metric that can be captured. Table 24 defines the requirements for instances of CIM_BaseMetricDefinition. This class is used as a basis for 10.6 to 10.10.

1130 **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.

1131 10.6 CIM_BaseMetricDefinition — Instantaneous Metric

1132 Table 25 describes the requirements for using CIM_BaseMetricDefinition to define an Instantaneous
1133 Metric. These constraints are in addition to those specified in 10.5.

1134 **Table 25 – Class: CIM_BaseMetricDefinition – Instantaneous Metric**

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

1135 10.7 CIM_BaseMetricDefinition — Interval Metric

1136 Table 26 describes the requirements for using CIM_BaseMetricDefinition to define an Interval Metric.
1137 These constraints are in addition to those specified in 10.5.

1138 **Table 26 – Class: CIM_BaseMetricDefinition – Interval Metric**

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 3 (Interval)

1139 10.8 CIM_BaseMetricDefinition — Startup Interval Metric

1140 Table 27 describes the requirements for using CIM_BaseMetricDefinition to define a Startup Interval
1141 Metric. These constraints are in addition to those specified in 10.5.

1142 **Table 27 – Class: CIM_BaseMetricDefinition – Startup Interval Metric**

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 4 (Startup Interval)

1143 10.9 CIM_BaseMetricDefinition — Summation Metric

1144 Table 28 describes the requirements for using CIM_BaseMetricDefinition to define a Summation Metric.
1145 These constraints are in addition to those specified in 10.5.

1146 **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)

1147 **10.10 CIM_BaseMetricDefinition — Current Data**

1148 Table 29 describes the requirements for using CIM_BaseMetricDefinition to define the metric to be used
 1149 with current data. These constraints are in addition to those specified in 10.5.

1150 **Table 29 – Class: CIM_BaseMetricDefinition – Current Data**

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

1151 **10.11 CIM_BaseMetricValue**

1152 CIM_BaseMetricValue conveys the actual recorded data of a metric. Table 30 describes the requirements
 1153 for instances of CIM_BaseMetricValue. This class is used as a basis for 10.12 to 10.16.

1154

1155 **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

1156 **10.12 CIM_BaseMetricValue — Current Data**

1157 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 31 describes the
 1158 requirements for using CIM_BaseMetricValue to report the metric for current data. These constraints are
 1159 in addition to those specified in 10.11.

1160 **Table 31 – Class: CIM_BaseMetricValue – Current Data**

Properties	Requirement	Notes
Timestamp	Mandatory	None
Volatile	Mandatory	Matches TRUE

1161 10.13 CIM_BaseMetricValue — Interval Metrics

1162 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 32 describes the
 1163 requirements for using CIM_BaseMetricValue to report the metric for interval metrics. These constraints
 1164 are in addition to those specified in 10.11.

1165 **Table 32 – Class: CIM_BaseMetricValue – Interval Metrics**

Properties	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None

1166 10.14 CIM_BaseMetricValue — Startup Interval Metrics

1167 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 33 describes the
 1168 requirements for using CIM_BaseMetricValue to report the metric for startup interval metrics. These
 1169 constraints are in addition to those specified in 10.11.

1170 **Table 33 – Class: CIM_BaseMetricValue – Startup Interval Metrics**

Properties	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None

1171 10.15 CIM_BaseMetricValue — Summation Metric

1172 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 34 describes the
 1173 requirements for using CIM_BaseMetricValue to report the metric for a Summation Metric. These
 1174 constraints are in addition to those specified in 10.11.

1175 **Table 34 – Class: CIM_BaseMetricValue – Summation Metric**

Properties	Requirement	Notes
Timestamp	Mandatory	None

1176 10.16 CIM_BaseMetricValue — Long-Term Monitoring

1177 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 35 describes the
 1178 requirements for using CIM_BaseMetricValue to report a metric for long-term monitoring. These
 1179 constraints are in addition to those specified in 10.11.

1180 **Table 35 – Class: CIM_BaseMetricValue – Long-Term Monitoring**

Properties	Requirement	Notes
Volatile	Mandatory	Matches FALSE

1181 **10.17 CIM_ConcreteDependency (Definition)**

1182 Table 36 details the requirements for instances of CIM_ConcreteDependency.

1183 **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

1184 **10.18 CIM_ConcreteDependency (Value)**

1185 Table 37 details the requirements for instances of CIM_ConcreteDependency.

1186 **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

1187 **10.19 CIM_ElementCapabilities**

1188 CIM_ElementCapabilities associates an instance of CIM_MetricServiceCapabilities with the Central
1189 Instance. Table 38 details the requirements for instances of CIM_ElementCapabilities.

1190 **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

1191 **10.20 CIM_HostedService**

1192 Table 39 details the requirements for instances of CIM_HostedService.

1193 **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..*

1194 **10.21 CIM_MetricDefForME**1195 CIM_MetricForME relates a metric to the managed element for which it was measured. Table 40 details
1196 the requirements for instances of CIM_MetricDefForME.1197 **Table 40 – Class: CIM_MetricDefForME**

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

1198 **10.22 CIM_MetricForME**1199 CIM_MetricForME relates a metric to the managed element for which it was measured. Table 41 details
1200 the requirements for instances of CIM_MetricForME.1201 **Table 41 – Class: CIM_MetricForME**

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

1202 **10.23 CIM_MetricInstance**1203 CIM_MetricInstance relates a CIM_BaseMetricValue to the CIM_BaseMetricDefinition that defines it.
1204 Table 42 details the requirements for instances of CIM_MetricInstance.1205 **Table 42 – Class: CIM_MetricInstance**

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

1206 **10.24 CIM_MetricService**

1207 Table 43 details the requirements for instances of CIM_MetricService.

1208 **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.

1209 **10.25 CIM_MetricServiceCapabilities**

1210 CIM_MetricServiceCapabilities indicates support for managing the state of the service as well as the
 1211 accounts with which the service is associated. Table 44 details the requirements for instances of
 1212 CIM_MetricServiceCapabilities.

1213 **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

1214 **10.26 CIM_RegisteredProfile**

1215 CIM_RegisteredProfile identifies the *Base Metrics Profile*. The CIM_RegisteredProfile class is defined by
 1216 the [Profile Registration Profile](#). With the exception of the mandatory values specified for the properties in
 1217 Table 45, the behavior of the CIM_RegisteredProfile instance is in accordance with the constraints
 1218 specified in the [Profile Registration Profile](#).

1219 **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 "1.0.1".
RegisteredOrganization	Mandatory	This property shall have a value of 2 (DMTF).

1220 **10.27 CIM_ServiceAffectsElement**

1221 CIM_ServiceAffectsElement is used to associate an instance of CIM_MetricService with an instance of
 1222 CIM_BaseMetricDefinition or CIM_AggregationMetricDefinition that represents a metric that could be
 1223 controlled using the service. Table 46 contains the requirements for elements of this class.

1224 **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)

1225

ANNEX A
(Informative)

Change Log

1226
1227
1228
1229
1230

Version	Date	Description
1.0.0	2009-06-16	DMTF Standard
1.0.1	2009-12-11	DMTF Standard, with the following changes: <ul style="list-style-type: none"> • Corrected inconsistencies based on the published profiles incorporating metric definitions and DSP0004 programmatic unit definitions.

1231

ANNEX B (Informative)

Guide for Common Metrics

1232
1233
1234
1235

1236 This annex provides an informative list of the combined mandatory properties for instances of
1237 CIM_BaseMetricDefinition, CIM_AggregationMetricDefinition, CIM_BaseMetricValue, and
1238 CIM_AggregationMetricValue if used to represent common metrics. Each of the data cells of the tables
1239 lists mandatory properties and their value formulations for a specific type of metric. Each table
1240 corresponds to a different type of metrics grouped by value formulation. The rows represent the different
1241 type of metrics based on the time scope that metric describes. The columns describe the different type of
1242 metrics based on the metric collection access type used.

1243 In order to determine the mandatory set of properties, match the type of metric to one of the data cells
1244 based on the metric's value formulation, time scope, and collection access type.

1245 The following conventions are used in the table:

- 1246 • BMD – the properties that follow are required on the instance of CIM_BaseMetricDefinition
- 1247 • BMV – the properties that follow are required on the instance of CIM_BaseMetricValue
- 1248 • AMD – the properties that follow are required on the instance of
1249 CIM_AggregationMetricDefinition
- 1250 • AMV – the properties that follow are required on the instance of CIM_AggregationMetricValue
- 1251 • A property name without a value specified is required, and the value is not fixed.
- 1252 • A property name followed by a value assignment is required with the specified value fixed.

1253 NOTE: If there is a mismatch between the mandatory set of properties and/or the properties' value formulation
1254 indicated by the tables in this annex and the requirements detailed in clauses 7 and 10, the requirements
1255 mandated in clauses 7 and 10 take precedence.

1256 B.1 Simple Metric

1257 Table B.1 describes the mandatory properties for simple metric as described in 6.3.1 according to the
1258 appropriate metric access type and time scope.

1259

Table B.1 – Simple Metric

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

	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 = TRUE InstanceID Duration Timestamp	BMV MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp

1260 **B.2 Summation Metric**

1261 Table B.2 describes the mandatory properties for summation metric as described in 6.3.2 according to the
 1262 appropriate metric access type and time scope.

1263 **Table B.2 – Summation Metric**

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

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Interval Metrics	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType = 3 (Periodic) or 2 (OnChange)</p> <p>Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType = 4 (OnRequest)</p> <p>Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits</p> <p>Name</p>
	<p>BMV</p> <p>MetricDefinitionId</p> <p>MetricValue</p> <p>Volatile = TRUE</p> <p>InstanceID</p> <p>Timestamp</p>	<p>BMV</p> <p>MetricDefinitionId</p> <p>MetricValue</p> <p>Volatile = TRUE</p> <p>InstanceID</p> <p>Timestamp</p>	<p>BMV</p> <p>MetricDefinitionId</p> <p>MetricValue</p> <p>Volatile = TRUE</p> <p>InstanceID</p> <p>Timestamp</p>	<p>BMV</p> <p>MetricDefinitionId</p> <p>MetricValue</p> <p>Volatile = FALSE</p> <p>InstanceID</p> <p>Timestamp</p>	<p>BMV</p> <p>MetricDefinitionId</p> <p>MetricValue</p> <p>Volatile</p> <p>InstanceID</p> <p>Timestamp</p>
Startup Metrics	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType = 3 (Periodic) or 2 (OnChange)</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType = 4 (OnRequest)</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>
	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType = 3 (Periodic) or 2 (OnChange)</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType = 4 (OnRequest)</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>	<p>BMD</p> <p>ChangeType = 3 (Counter)</p> <p>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)</p> <p>ElementName</p> <p>GatheringType</p> <p>Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits</p> <p>Name</p>

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

1264 **B.3 Aggregation Metric**

1265 Table B.3 describes the mandatory properties for aggregation metric as described in section 6.3.3
 1266 according to the appropriate metric access type and time scope.

1267 **Table B.3 – Aggregation Metric**

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

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

	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
	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp
	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration

1268 **B.4 Aggregation Metric — Low Watermark**

1269 Table B.4 describes the mandatory properties for low watermark as a type of an aggregation metric as
 1270 described in section 6.3.3.1 according to the appropriate metric access type and time scope.

1271 **Table B.4 – Aggregation Metric – Low Watermark**

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

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

	Current Data	Current Data – Online Monitoring	Current Data – Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Startup Metrics	AMD	AMD	AMD	AMD	AMD
	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)
	DataType	DataType	DataType	DataType	DataType
	ElementName	ElementName	ElementName	ElementName	ElementName
	GatheringType	GatheringType = 3 (Periodic) or 2 (OnChange)	GatheringType = 4 (OnRequest)	GatheringType	GatheringType
	Id	Id	Id	Id	Id
	TimeScope = 4 (StartupInterval)	Id	TimeScope = 4 (StartupInterval)	TimeScope = 4 (StartupInterval)	TimeScope = 4 (StartupInterval)
	ProgrammaticUnits	TimeScope = 4 (StartupInterval)	ProgrammaticUnits	ProgrammaticUnits	ProgrammaticUnits
	Name	ProgrammaticUnits	Name	Name	Name
	SimpleFunction = 2 (Minimum)	Name	SimpleFunction = 2 (Minimum)	SimpleFunction = 2 (Minimum)	SimpleFunction = 2 (Minimum)
		SimpleFunction = 2 (Minimum)			
	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	
AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	
AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	

1272 **B.5 Aggregation Metric — High Watermark**

1273 Table B.5 describes the mandatory properties for high watermark as a type of an aggregation metric as
 1274 described in section 6.3.3.1 according to the appropriate metric access type and time scope.

1275 **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	AMD	AMD	AMD	AMD
	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)	ChangeType = 5 (Simple Function)
	DataType	DataType	DataType	DataType	DataType
	ElementName	ElementName	ElementName	ElementName	ElementName
	GatheringType	GatheringType = 3 (Periodic) or 2 (OnChange)	GatheringType = 4 (OnRequest)	GatheringType	GatheringType
	Id	Id	Id	Id	Id
	IsContinuous = TRUE	Id	IsContinuous = TRUE	IsContinuous = TRUE	IsContinuous = TRUE
	TimeScope = 2 (Point)	IsContinuous = TRUE	TimeScope = 2 (Point)	TimeScope = 2 (Point)	TimeScope = 2 (Point)
	ProgrammaticUnits	TimeScope = 2 (Point)	ProgrammaticUnits	ProgrammaticUnits	ProgrammaticUnits
	Name	ProgrammaticUnits	Name	Name	Name
	SimpleFunction = 3 (Maximum)	Name	SimpleFunction = 3 (Maximum)	SimpleFunction = 3 (Maximum)	SimpleFunction = 3 (Maximum)
		SimpleFunction = 3 (Maximum)			
	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	AggregationTime-Stamp	AggregationTime-Stamp	
Timestamp	Timestamp	Timestamp	AggregationDuration	AggregationDuration	
AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp			
AggregationDuration	AggregationDuration	AggregationDuration			

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

	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
	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp	AggregationTime-Stamp
	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration

1276

1277

1278

Bibliography

1279 DMTF DSP1073, *Capacity Metrics Profile 1.0*

1280