



1

2

3

4

Document Number: DSP1053

Date: 2009-06-16

Version: 1.0.0

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26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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	2.1 Approved References	9
38	2.2 Other References.....	9
39	3 Terms and Definitions	9
40	4 Symbols and Abbreviated Terms	12
41	5 Synopsis.....	12
42	6 Description (Informative)	12
43	6.1 Metric Access Types.....	13
44	6.2 Metric Time Scope.....	14
45	6.3 Metric Value Formulation.....	15
46	6.4 Metric Context.....	16
47	7 Implementation.....	16
48	7.1 Common Requirements	16
49	7.2 Modeling Metric Access Types	18
50	7.3 Modeling Metric Time Scope	18
51	7.4 Modeling Metric Value Formulation	19
52	7.5 Relationship between Aggregation and Base Metrics	19
53	7.6 Constraints on Metric Values for Controllable Metrics.....	20
54	8 Methods.....	20
55	8.1 CIM_MetricService.ShowMetrics()	20
56	8.2 CIM_MetricService.ShowMetricsByClass()	21
57	8.3 CIM_MetricService.ControlMetrics().....	22
58	8.4 CIM_MetricService.ControlMetricsByClass()	24
59	8.5 CIM_MetricService.GetMetricValues()	25
60	8.6 Profile Conventions for Operations.....	25
61	8.7 CIM_AggregationMetricDefinition	26
62	8.8 CIM_AggregationMetricValue	26
63	8.9 CIM_BaseMetricDefinition	26
64	8.10 CIM_BaseMetricValue	26
65	8.11 CIM_ConcreteDependency	26
66	8.12 CIM_ElementCapabilities	26
67	8.13 CIM_HostedService	27
68	8.14 CIM_MetricDefForME	27
69	8.15 CIM_MetricForME.....	27
70	8.16 CIM_MetricInstance	28
71	8.17 CIM_MetricService.....	28
72	8.18 CIM_MetricServiceCapabilities	28
73	8.19 CIM_ServiceAffectsElement	28
74	9 Use Cases (Informative).....	29
75	9.1 Instructions Executed per Second	29
76	9.2 Object Diagram for Startup Interval Time Scope.....	33
77	9.3 Metric Definition for Multiple Instances of CIM_ManagedElement.....	34
78	9.4 Controllable Metrics	35
79	9.5 Aggregation Metrics	41
80	9.6 Metric Context.....	43
81	9.7 Find All Metric Definitions for a Managed Element.....	44
82	9.8 Find the Metric Value for a Managed Element	44
83	9.9 Find a Standard Metric for a Managed Element.....	44

84	9.10 Retrieve a Metric Value.....	44
85	9.11 Find All Metrics Available for a Managed Element within an Enumeration Scope.....	45
86	9.12 Find All Metrics Available within an Enumeration Scope for All Instances of a CIM Class.....	45
87	9.13 Determine whether a Metric Can Be Discretely Controlled for a Specific Managed	
88	Element.....	46
89	9.14 Enable a Specific Metric for a Specific Managed Element.....	46
90	9.15 Find All Managed Elements within an Enumeration Scope for which a Metric Is Currently	
91	Being Collected.....	46
92	10 CIM Elements.....	47
93	10.1 CIM_AggregationMetricDefinition.....	48
94	10.2 CIM_AggregationMetricDefinition (Low Watermark).....	48
95	10.3 CIM_AggregationMetricDefinition (High Watermark).....	49
96	10.4 CIM_AggregationMetricValue.....	49
97	10.5 CIM_BaseMetricDefinition.....	50
98	10.6 CIM_BaseMetricDefinition — Instantaneous Metric.....	50
99	10.7 CIM_BaseMetricDefinition — Interval Metric.....	50
100	10.8 CIM_BaseMetricDefinition — Startup Interval Metric.....	51
101	10.9 CIM_BaseMetricDefinition — Summation Metric.....	51
102	10.10 CIM_BaseMetricDefinition — Current Data.....	51
103	10.11 CIM_BaseMetricValue.....	52
104	10.12 CIM_BaseMetricValue — Current Data.....	52
105	10.13 CIM_BaseMetricValue — Interval Metrics.....	52
106	10.14 CIM_BaseMetricValue — Startup Interval Metrics.....	53
107	10.15 CIM_BaseMetricValue — Summation Metric.....	53
108	10.16 CIM_BaseMetricValue — Long-Term Monitoring.....	53
109	10.17 CIM_ConcreteDependency (Definition).....	53
110	10.18 CIM_ConcreteDependency (Value).....	54
111	10.19 CIM_ElementCapabilities.....	54
112	10.20 CIM_HostedService.....	54
113	10.21 CIM_MetricDefForME.....	55
114	10.22 CIM_MetricForME.....	55
115	10.23 CIM_MetricInstance.....	55
116	10.24 CIM_MetricService.....	56
117	10.25 CIM_MetricServiceCapabilities.....	56
118	10.26 CIM_RegisteredProfile.....	57
119	10.27 CIM_ServiceAffectsElement.....	57
120	ANNEX A (Informative) Change Log.....	58
121	ANNEX B (Informative) Guide for Common Metrics.....	59
122	Bibliography.....	72
123		
124	Figures	
125	Figure 1 – Base Metrics Profile: Class Diagram.....	13
126	Figure 2 – Interval Metrics.....	30
127	Figure 3 – Instantaneous Counter.....	31
128	Figure 4 – Instantaneous Gauge.....	32
129	Figure 5 – Usage Example for Startup Interval Time Scope.....	33
130	Figure 6 – Common Metric Definition for Multiple Instances of CIM_ManagedElement.....	34
131	Figure 7 – Advertising Support for Discrete Controllable Metrics.....	35
132	Figure 8 – Discrete Controllable Metrics (Before Enable).....	36
133	Figure 9 – Discrete Controllable Metrics (After Enable).....	37
134	Figure 10 – Bulk Controllable Metrics by Definition.....	38
135	Figure 11 – Bulk Controllable Metrics by Managed Element.....	39

136 Figure 12 – Bulk Controllable Metrics by Class 40

137 Figure 13 – Aggregation Metric without Base 41

138 Figure 14 – Aggregation Metric with Base 42

139 Figure 15 – Metric Context..... 43

140

141 **Tables**

142 Table 1 – Referenced Profiles 12

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

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

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

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

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

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

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

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

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

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

153 Table 12 – Operations: CIM_ConcreteDependency 26

154 Table 13 – Operations: CIM_ElementCapabilities 27

155 Table 14 – Operations: CIM_HostedService 27

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

157 Table 16 – Operations: CIM_MetricForME 28

158 Table 17 – Operations: CIM_MetricInstance 28

159 Table 18 – Operations: CIM_ServiceAffectsElement 29

160 Table 19 – CIM Elements: Base Metrics Profile 47

161 Table 20 – Class: CIM_AggregationMetricDefinition 48

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

163 Table 22 – Class: CIM_AggregationMetricDefinition (High Watermark) 49

164 Table 23 – Class: CIM_AggregationMetricValue 49

165 Table 24 – Class: CIM_BaseMetricDefinition 50

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

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

168 Table 27 – Class: CIM_BaseMetricDefinition – Startup Interval Metric..... 51

169 Table 28 – Class: CIM_BaseMetricDefinition – Summation Metric 51

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

171 Table 30 – Class: CIM_BaseMetricValue 52

172 Table 31 – Class: CIM_BaseMetricValue – Current Data 52

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

174 Table 33 – Class: CIM_BaseMetricValue – Startup Interval Metrics 53

175 Table 34 – Class: CIM_BaseMetricValue – Summation Metric 53

176 Table 35 – Class: CIM_BaseMetricValue – Long-Term Monitoring..... 53

177 Table 36 – Class: CIM_ConcreteDependency (Definition) 53

178 Table 37 – Class: CIM_ConcreteDependency (Value) 54

179 Table 38 – Class: CIM_ElementCapabilities..... 54

180 Table 39 – Class: CIM_HostedService 54

181 Table 40 – Class: CIM_MetricDefForME 55

182 Table 41 – Class: CIM_MetricForME..... 55

183	Table 42 – Class: CIM_MetricInstance	55
184	Table 43 – Class: CIM_MetricService.....	56
185	Table 44 – Class: CIM_MetricServiceCapabilities	56
186	Table 45 – Class: CIM_RegisteredProfile.....	57
187	Table 46 – Class: CIM_ServiceAffectsElement	57
188	Table B.1 – Simple Metric	60
189	Table B.2 – Summation Metric.....	62
190	Table B.3 – Aggregation Metric	64
191	Table B.4 – Aggregation Metric – Low Watermark	66
192	Table B.5 – Aggregation Metric – High Watermark	69
193		

194

Foreword

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

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

198 **Acknowledgments**

199 The authors wish to acknowledge the following people.

200 **Editors:**

- 201 • Oliver Benke – IBM
- 202 • Aaron Merkin – IBM

203 **Contributors:**

- 204 • Karl Schopmeyer – The Open Group
- 205 • Andreas Maier – IBM
- 206 • Khachatur Papanyan – Dell

207

208

Introduction

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

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

215

Base Metrics Profile

216 1 Scope

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

221 2 Normative References

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

225 2.1 Approved References

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

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

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

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

234 2.2 Other References

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

237 3 Terms and Definitions

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

239 3.1

240 **can**

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

242 3.2

243 **cannot**

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

245 3.3

246 **conditional**

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

- 249 **3.4**
250 **mandatory**
251 indicates requirements to be followed strictly in order to conform to the document and from which no
252 deviation is permitted
- 253 **3.5**
254 **may**
255 indicates a course of action permissible within the limits of the document
- 256 **3.6**
257 **need not**
258 indicates a course of action permissible within the limits of the document
- 259 **3.7**
260 **optional**
261 indicates a course of action permissible within the limits of the document
- 262 **3.8**
263 **referencing profile**
264 indicates a profile that owns the definition of this class and can include a reference to this profile in its
265 "Referenced Profiles" table
- 266 **3.9**
267 **shall**
268 indicates requirements to be followed strictly in order to conform to the document and from which no
269 deviation is permitted
- 270 **3.10**
271 **shall not**
272 indicates requirements to be followed in order to conform to the document and from which no deviation is
273 permitted
- 274 **3.11**
275 **should**
276 indicates that among several possibilities, one is recommended as particularly suitable, without
277 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required
- 278 **3.12**
279 **should not**
280 indicates that a certain possibility or course of action is deprecated but not prohibited
- 281 **3.13**
282 **unspecified**
283 indicates that this profile does not define any constraints for the referenced CIM element or operation
- 284 **3.14**
285 **aggregation metric**
286 a type of metric that is derived by applying a formula or filter to a set of base metric values
- 287 **3.15**
288 **base metric**
289 a metric provided directly without a dependency on other metric values

- 290 **3.16**
291 **measured resource**
292 a managed object being measured, which is the resource to which base metric value instances are
293 associated
- 294 **3.17**
295 **sampling interval**
296 a value that determines how often new metric values are retrieved, if metrics are retrieved periodically
- 297 **3.18**
298 **current data**
299 the most current data available for a given metric. Online monitoring (3.19) and snapshot monitoring
300 (3.20) are types of current data access.
- 301 **3.19**
302 **online monitoring**
303 the process in which metric values (typically interval metrics) are gathered asynchronously to a request
304 from the instrumentation or reporting layer
- 305 **3.20**
306 **snapshot monitoring**
307 the process in which metric values are gathered synchronously with a request from the instrumentation or
308 reporting layer
- 309 **3.21**
310 **long-term monitoring**
311 the process in which metric values are captured during an interval
- 312 **3.22**
313 **event-based monitoring**
314 the process in which threshold values for metrics are used to trigger asynchronous notification
- 315 **3.23**
316 **instantaneous metrics**
317 metrics that apply to a particular point in time. An example of an instantaneous metric is the amount of
318 memory currently allocated to a virtual server.
- 319 **3.24**
320 **interval metrics**
321 metrics that apply to a time interval. An example of an interval metric is the average CPU utilization of a
322 server over the past hour.
- 323 **3.25**
324 **summation metrics**
325 a type of counter metric that reflects the accumulation of a value
- 326 **3.26**
327 **watermark metrics**
328 a type of aggregation metric used to capture the minimum or maximum value recorded for a monitored
329 value

330 4 Symbols and Abbreviated Terms

331 4.1

332 CPU

333 central processing unit

334 4.2

335 IEPS

336 instructions executed per second

337 4.3

338 UTC

339 Universal Time Coordinated

340 4.4

341 UUID

342 Universally Unique Identifier

343 5 Synopsis

344 **Profile Name:** Base Metrics

345 **Version:** 1.0.0

346 **Organization:** DMTF

347 **CIM Schema Version:** 2.22

348 **Central Class:** CIM_MetricService

349 **Scoping Class:** CIM_System

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

351

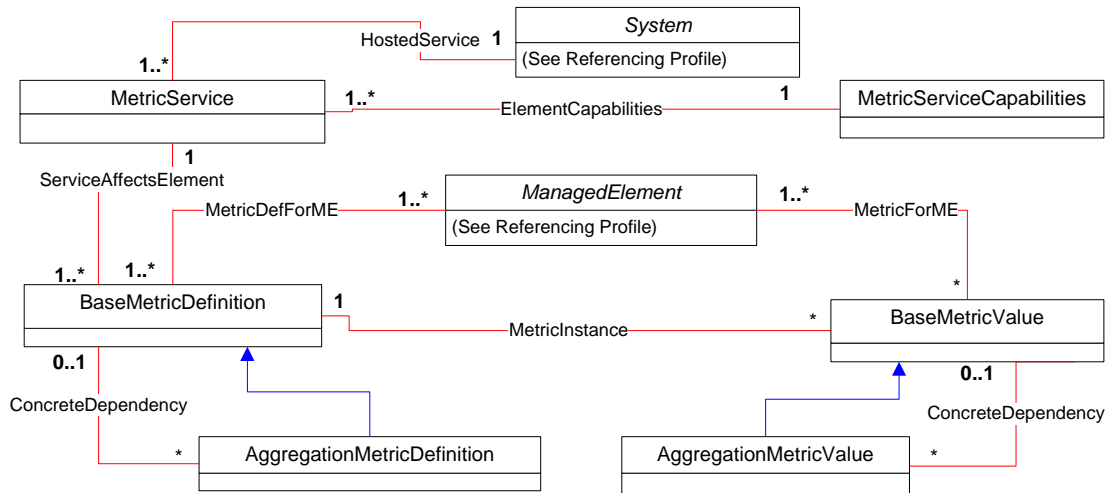
Table 1 – Referenced Profiles

Profile Name	Organization	Version	Relationship	Behavior
Profile Registration	DMTF	1.0	Mandatory	

352 6 Description (Informative)

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

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



356

357

Figure 1 – Base Metrics Profile: Class Diagram

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

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

- 366 • Metric access type
- 367 • Time scope of the metric
- 368 • Formulation of the metric value
- 369 • Metric context

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

374 6.1 Metric Access Types

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

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

380 6.1.1 Current Data

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

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

386 **6.1.1.1 Online Monitoring**

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

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

394 **6.1.1.2 Snapshot Monitoring**

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

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

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

403 **6.1.2 Long-Term Monitoring**

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

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

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

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

411 **6.1.3 Event-Based Monitoring**

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

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

417 **6.2 Metric Time Scope**

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

422 **6.2.1 Instantaneous Metrics**

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

426 **6.2.2 Interval Metrics**

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

431 **6.2.3 Startup Interval Metrics**

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

436 **6.3 Metric Value Formulation**

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

440 **6.3.1 Simple Metrics**

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

444 **6.3.2 Summation Metrics**

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

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

449 **6.3.3 Aggregation Metrics**

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

460 6.3.3.1 Watermark Metrics

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

464 6.4 Metric Context

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

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

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

483 7 Implementation

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

486 7.1 Common Requirements

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

490 7.1.1 Service and Capabilities

491 At least one instance of CIM_MetricService shall exist. Each instance of CIM_MetricService shall be
492 associated with exactly one instance of CIM_System through the CIM_HostedService association. Each
493 instance of CIM_MetricService shall be associated with exactly one instance of
494 CIM_MetricServiceCapabilities through the CIM_ElementCapabilities association. Each instance of
495 CIM_BaseMetricDefinition shall be associated with exactly one instance of CIM_MetricService through
496 the CIM_ServiceAffectsElement association.

497 7.1.2 Relating a Metric Definition and Metric Value

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

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

502 7.1.3 Identifying a Metric Definition

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

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

508 "CIM:DSP" <DSP#> ":" <unique identifier>

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

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

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

520 7.1.4 Identifying Metric Context

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

522 7.1.4.1 General Requirements

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

528 7.1.4.2 Breakdown Dimensions (Optional)

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

534 The CIM_BaseMetricDefinition.BreakdownDimensions property shall not be NULL.

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

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

540 <schemaName>_"<simpleClassName>

541 as specified in [DSP0004](#).

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

545 "CIM:DSP" <DSP#> ":" <unique identifier>

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

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

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

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

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

564 7.1.5 Gathering Type

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

569 7.2 Modeling Metric Access Types

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

573 7.2.1 Modeling Current Data Access Type (Optional)

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

577 7.3 Modeling Metric Time Scope

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

581 **7.3.1 Modeling Instantaneous Metrics (Optional)**

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

585 **7.3.2 Modeling Interval Metrics (Optional)**

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

588 **7.3.3 Modeling Interval Metrics (Optional)**

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

591 **7.4 Modeling Metric Value Formulation**

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

593 **7.4.1 Modeling Summation Metrics (Optional)**

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

596 **7.4.2 Modeling Aggregation Metrics (Optional)**

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

601 **7.4.2.1 Modeling Low Watermark Metrics (Optional)**

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

604 **7.4.2.2 Modeling High Watermark Metrics (Optional)**

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

607 **7.5 Relationship between Aggregation and Base Metrics**

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

617 7.6 Constraints on Metric Values for Controllable Metrics

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

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

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

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

636 8 Methods

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

641 8.1 CIM_MetricService.ShowMetrics()

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

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

647 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

649

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

650 **8.1.1 CIM_MetricService.ShowMetrics() Conditional Support**

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

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

657 **8.2 CIM_MetricService.ShowMetricsByClass()**

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

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

663 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

665

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

666 8.2.1 CIM_MetricService.ShowMetricsByClass() Conditional Support

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

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

673 8.3 CIM_MetricService.ControlMetrics()

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

- 675 • a metric for all instances of CIM_ManagedElement
- 676 • all metrics for a single CIM_ManagedElement instance
- 677 • a single metric for a single CIM_ManagedElement instance

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

681 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

683

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

684 **8.3.1 CIM_MetricService.ControlMetrics() Conditional Support**

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

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

691 **8.3.2 Parameter Validation**

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

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

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

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

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

724 8.4 CIM_MetricService.ControlMetricsByClass()

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

- 726 • a metric for all instances of a specific CIM class
- 727 • all metrics for all instances of a specific CIM class
- 728 • a single metric for a single CIM_ManagedElement

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

732 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

734 **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

735 8.4.1 CIM_MetricService.ControlMetricsByClass() Conditional Support

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

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

742 8.4.2 Parameter Validation

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

748 **8.5 CIM_MetricService.GetMetricValues()**

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

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

753 No standard messages are defined for this method.

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

Value	Description
0	Operation completed successfully
1	Operation unsupported
2	Failed

755 **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

756 **8.5.1 CIM_MetricService.GetMetricValues() Conditional Support**

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

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

763 **8.6 Profile Conventions for Operations**

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

766 The default list of operations is as follows:

- 767 • GetInstance
- 768 • Associators
- 769 • AssociatorNames
- 770 • References

- 771 • ReferenceNames
- 772 • EnumerateInstances
- 773 • EnumerateInstanceNames

774 8.7 CIM_AggregationMetricDefinition

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 8.8 CIM_AggregationMetricValue

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

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

780 8.9 CIM_BaseMetricDefinition

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

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

783 8.10 CIM_BaseMetricValue

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

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

786 8.11 CIM_ConcreteDependency

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

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

791 **Table 12 – Operations: CIM_ConcreteDependency**

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

792 8.12 CIM_ElementCapabilities

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

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

797

Table 13 – Operations: CIM_ElementCapabilities

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

798 **8.13 CIM_HostedService**

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

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

803

Table 14 – Operations: CIM_HostedService

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

804 **8.14 CIM_MetricDefForME**

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

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

809

Table 15 – Operations: CIM_MetricDefForME

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

810 **8.15 CIM_MetricForME**

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

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

815

Table 16 – Operations: CIM_MetricForME

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

816 8.16 CIM_MetricInstance

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

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

821

Table 17 – Operations: CIM_MetricInstance

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

822 8.17 CIM_MetricService

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

824 8.18 CIM_MetricServiceCapabilities

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

826 8.19 CIM_ServiceAffectsElement

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

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

831

Table 18 – Operations: CIM_ServiceAffectsElement

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

832 **9 Use Cases (Informative)**

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

834 **9.1 Instructions Executed per Second**

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

838 **9.1.1 Interval Metrics**

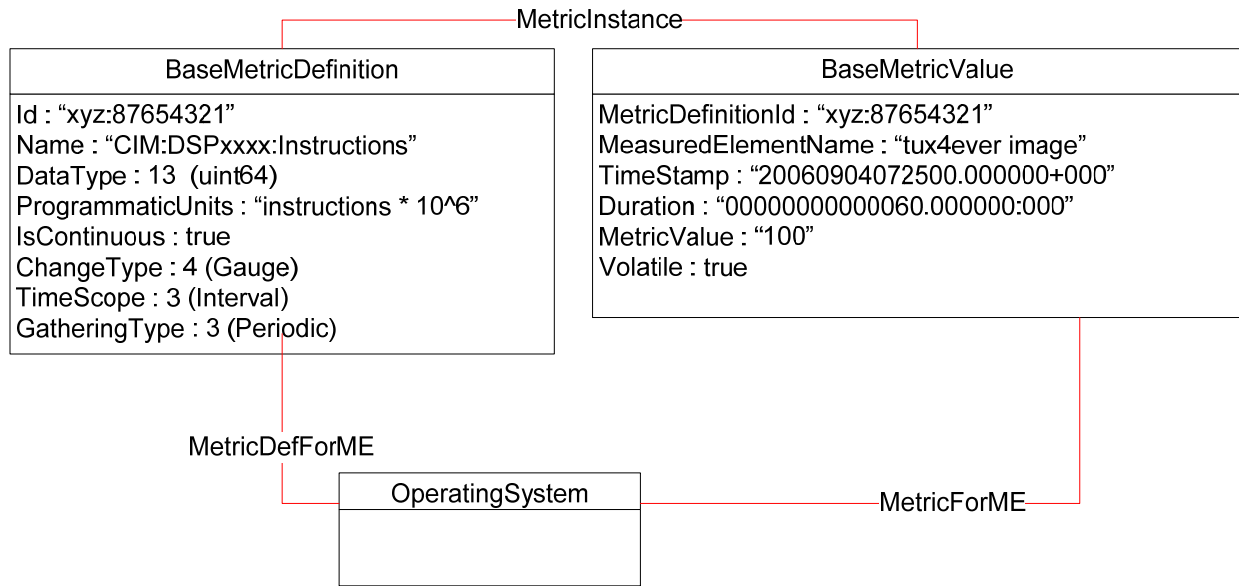
839 Figure 2 presents an object diagram for an implementation of an interval metric showing the instructions
 840 executed per second for an operating system image. There is one instance of the BaseMetricValue class
 841 with a TimeStamp property value of 07:25:00 A.M. at 9/4/2006, a Duration property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

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

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

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

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



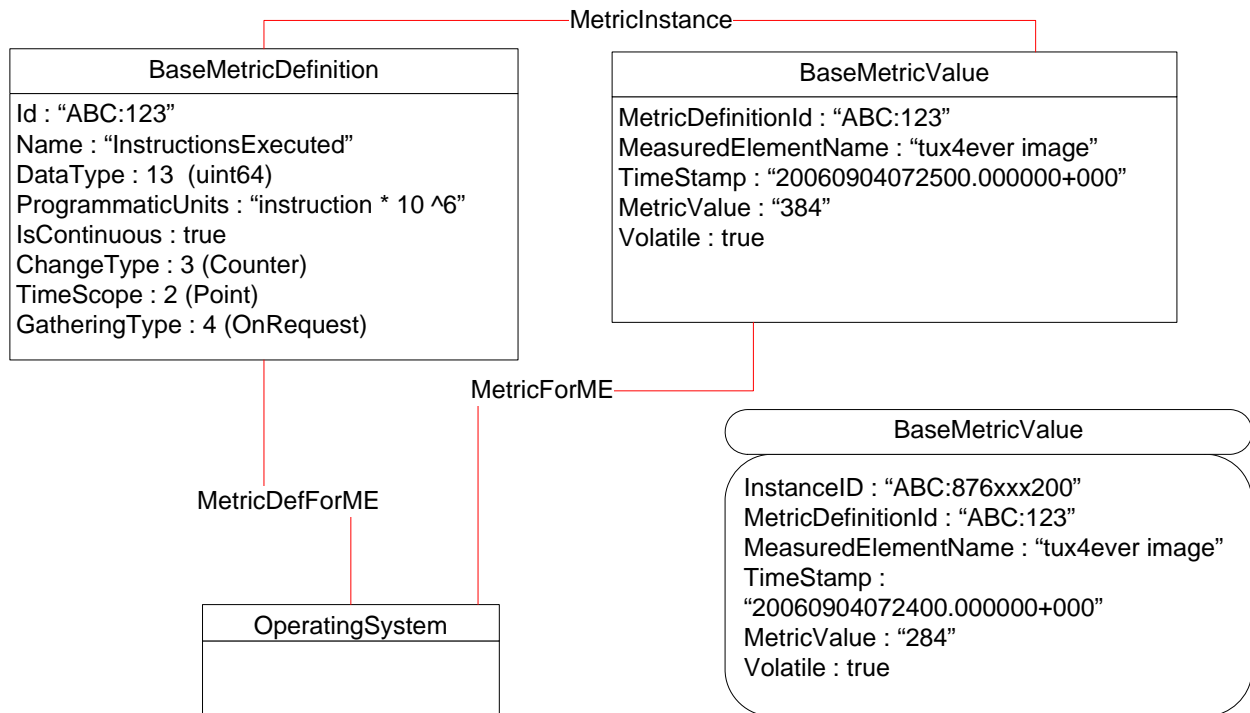
856

857

Figure 2 – Interval Metrics

858 **9.1.2 Instantaneous Counter**

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



866

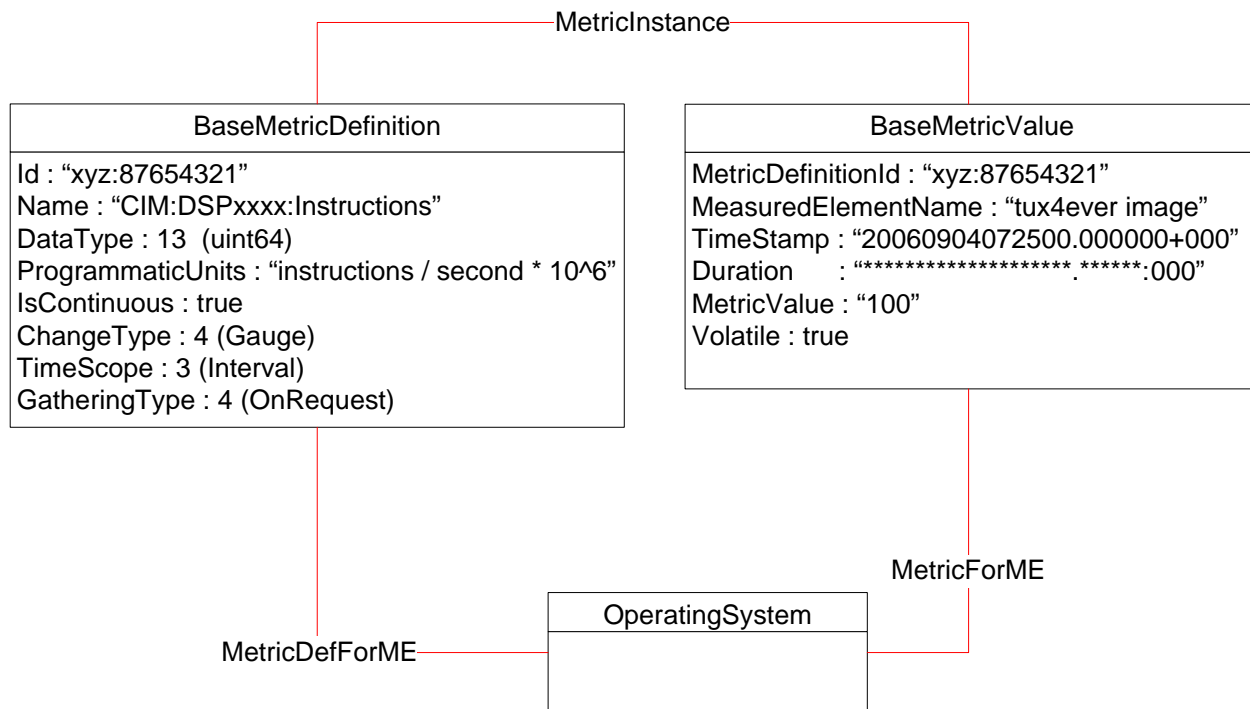
867

Figure 3 – Instantaneous Counter

868 **9.1.3 Instantaneous Gauge**

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

873 The ProgrammaticUnits property indicates that the metric reports millions of instructions per second. The
 874 CIM_BaseMetricDefinition.TimeScope property indicates that the metric is an interval metric. The
 875 CIM_BaseMetricValue.Duration property indicates that there is no precision to the reported interval
 876 duration. The current values of the properties of the CIM_BaseMetricValue instance indicate that, as of
 877 07:25:00 A.M. at 9/4/2006, an average of 100 million instructions were executed per second.



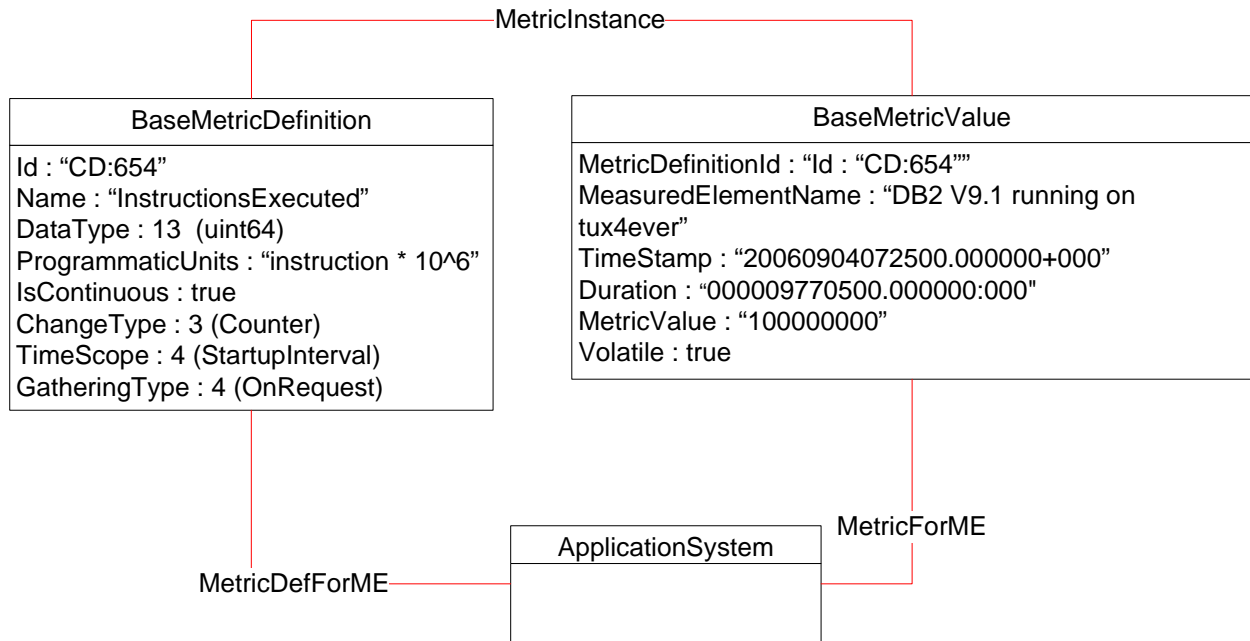
878

879

Figure 4 – Instantaneous Gauge

880 **9.2 Object Diagram for Startup Interval Time Scope**

881 A value of 4 (StartupInterval) for the TimeScope property indicates that the metric applies to an interval
 882 that began at the startup of the measured resource. The example in Figure 5 shows that at 07:25:00 A.M.
 883 on 09/04/2006, the associated application system "DB2 V9.1 on tux4ever" was running for a duration of
 884 977 days and 5 hours, consuming 100 million resources. The associated metric is "InstructionsExecuted",
 885 with a unit of "Million Instructions".



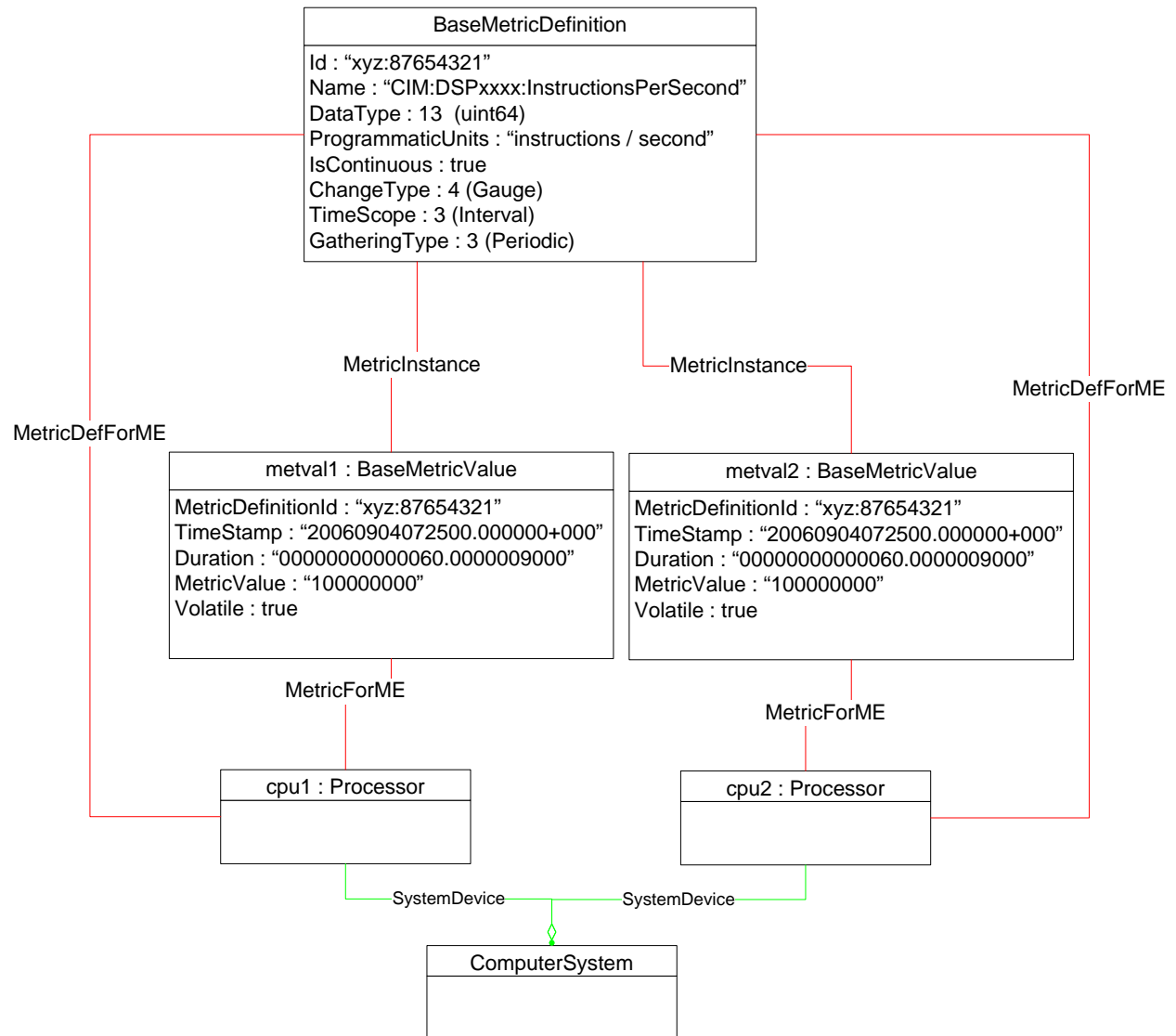
886

887

Figure 5 – Usage Example for Startup Interval Time Scope

888 **9.3 Metric Definition for Multiple Instances of CIM_ManagedElement**

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

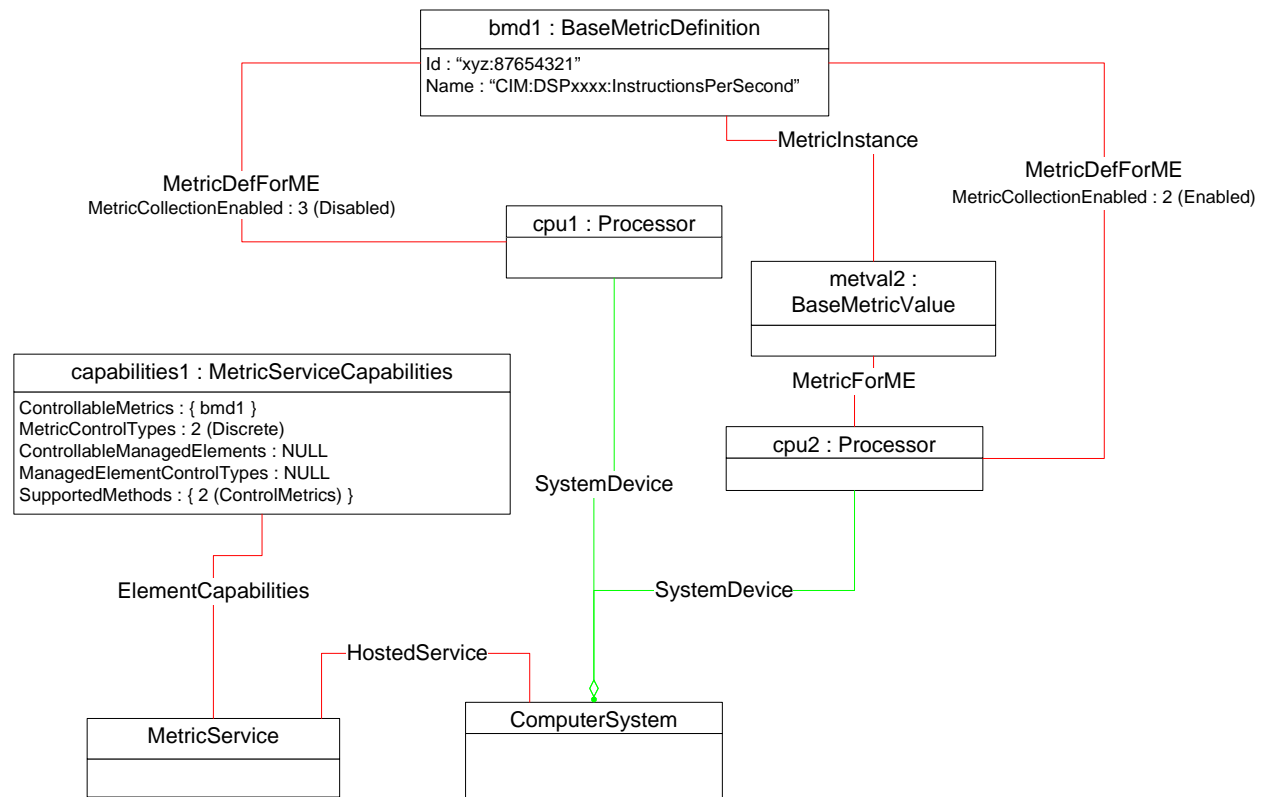


892

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

894 **9.4 Controllable Metrics**

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



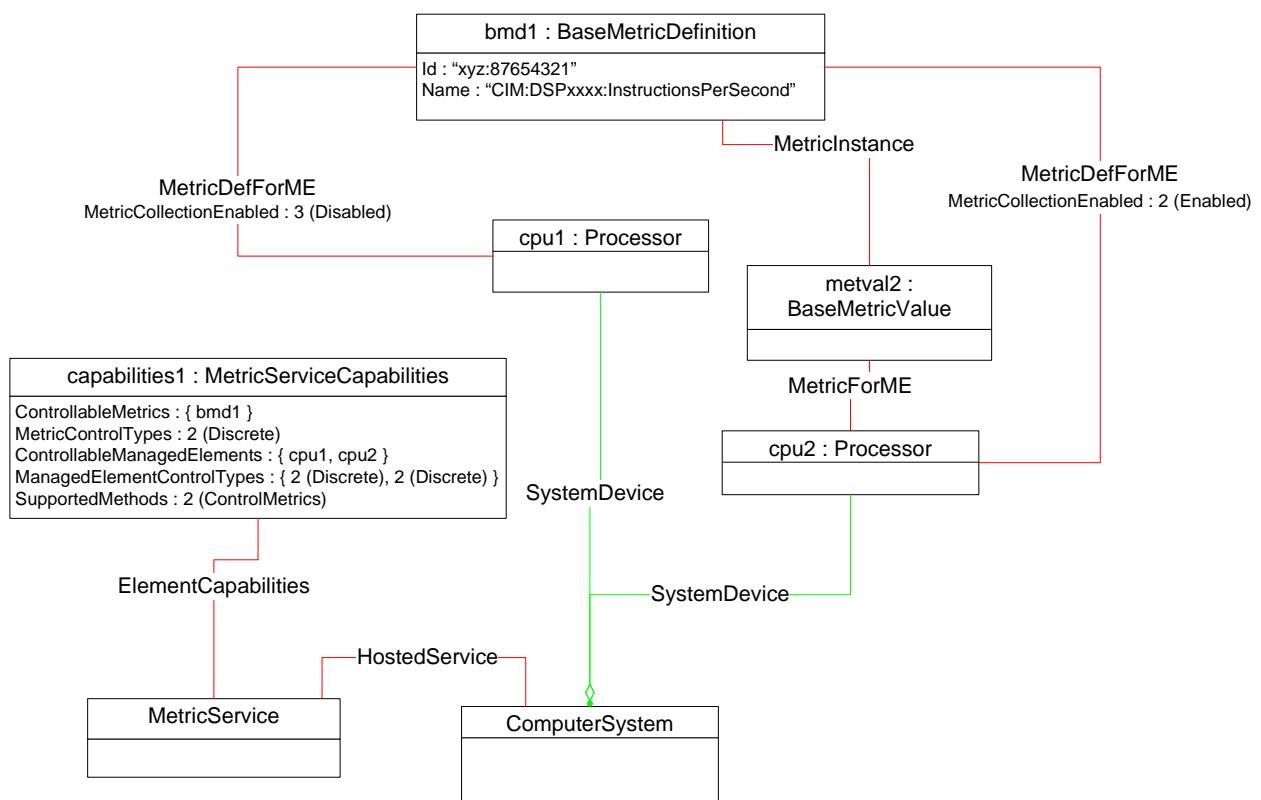
906

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

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

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

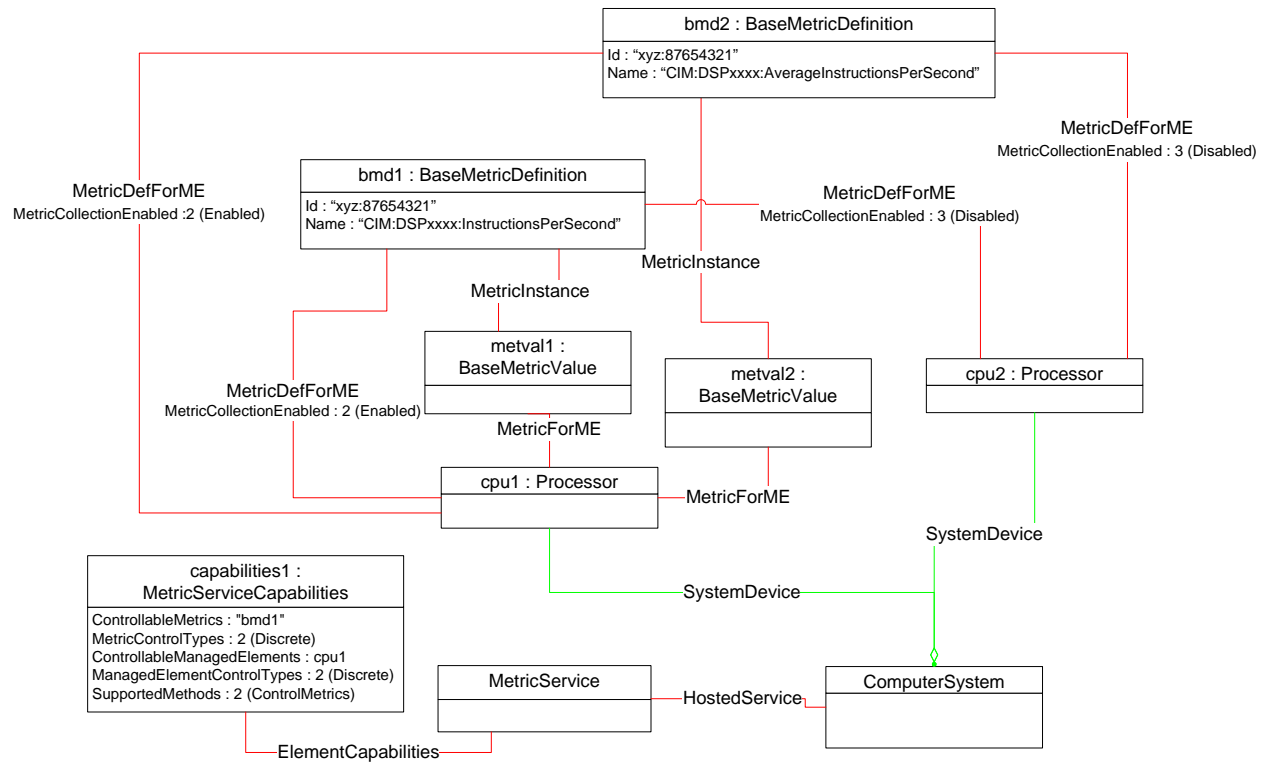
918 In the object diagram shown in Figure 8, collection of the metric for cpu1 has been disabled. This is
 919 indicated by the value of the MetricCollectionEnabled property of the instance of CIM_MetricDefForME
 920 that associates bmd1 with cpu1. The CIM_ServiceAffectsElement associations between the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



922

923 **Figure 8 – Discrete Controllable Metrics (Before Enable)**

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

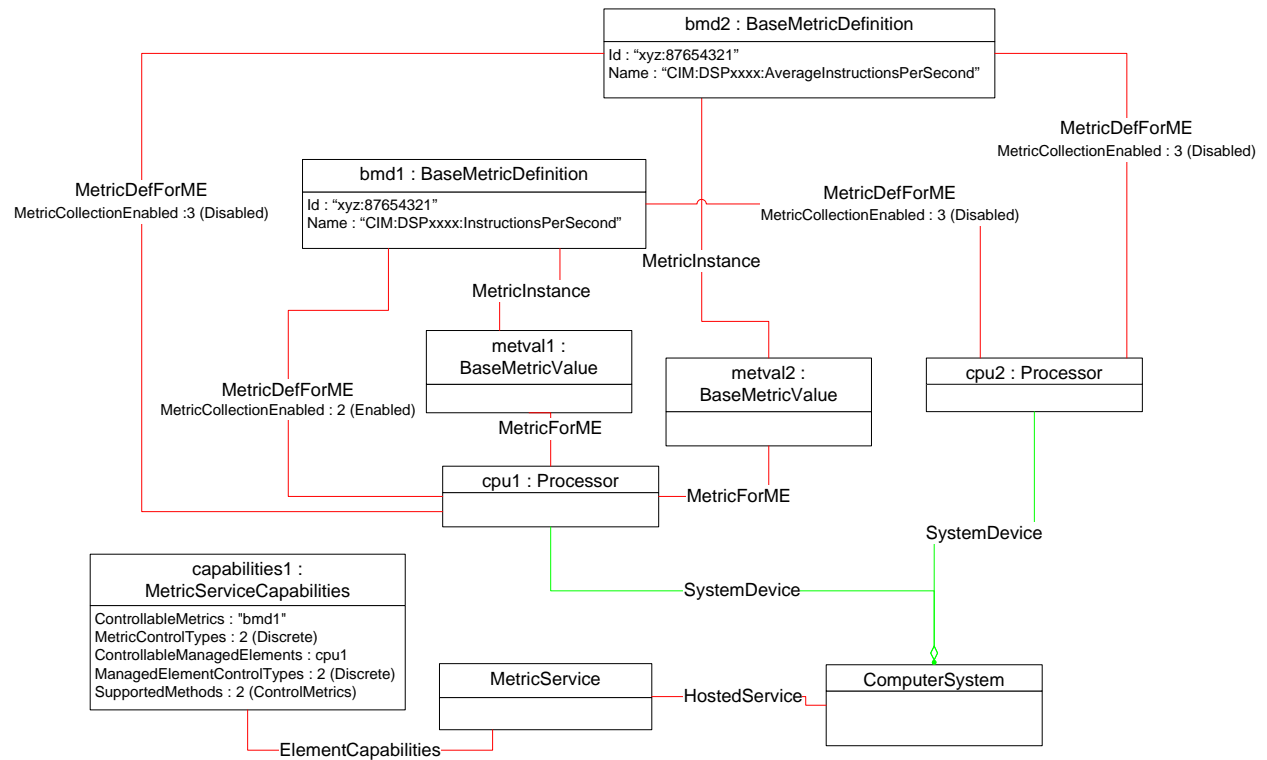


928

929

Figure 9 – Discrete Controllable Metrics (After Enable)

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

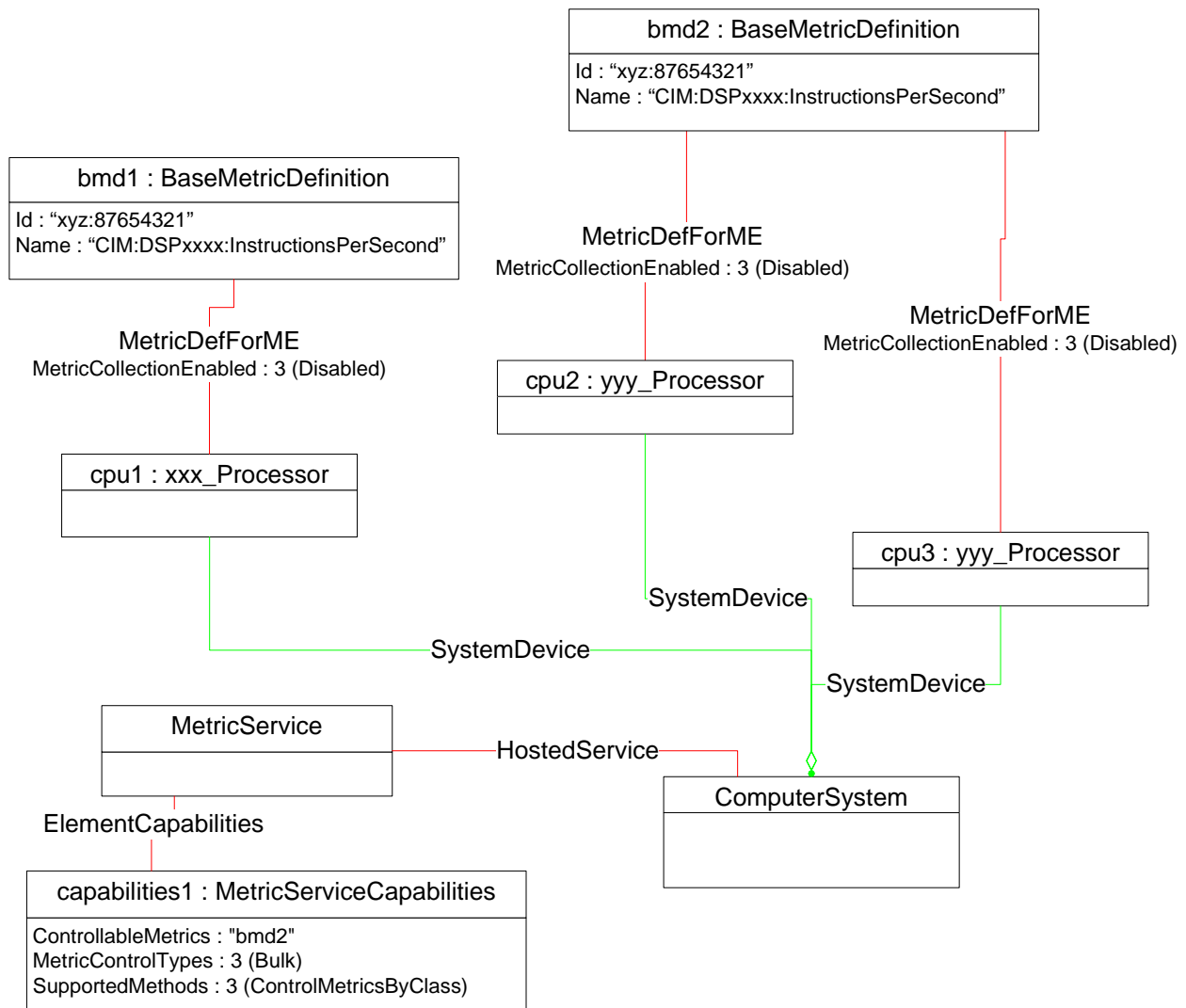


939

940

Figure 10 – Bulk Controllable Metrics by Definition

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

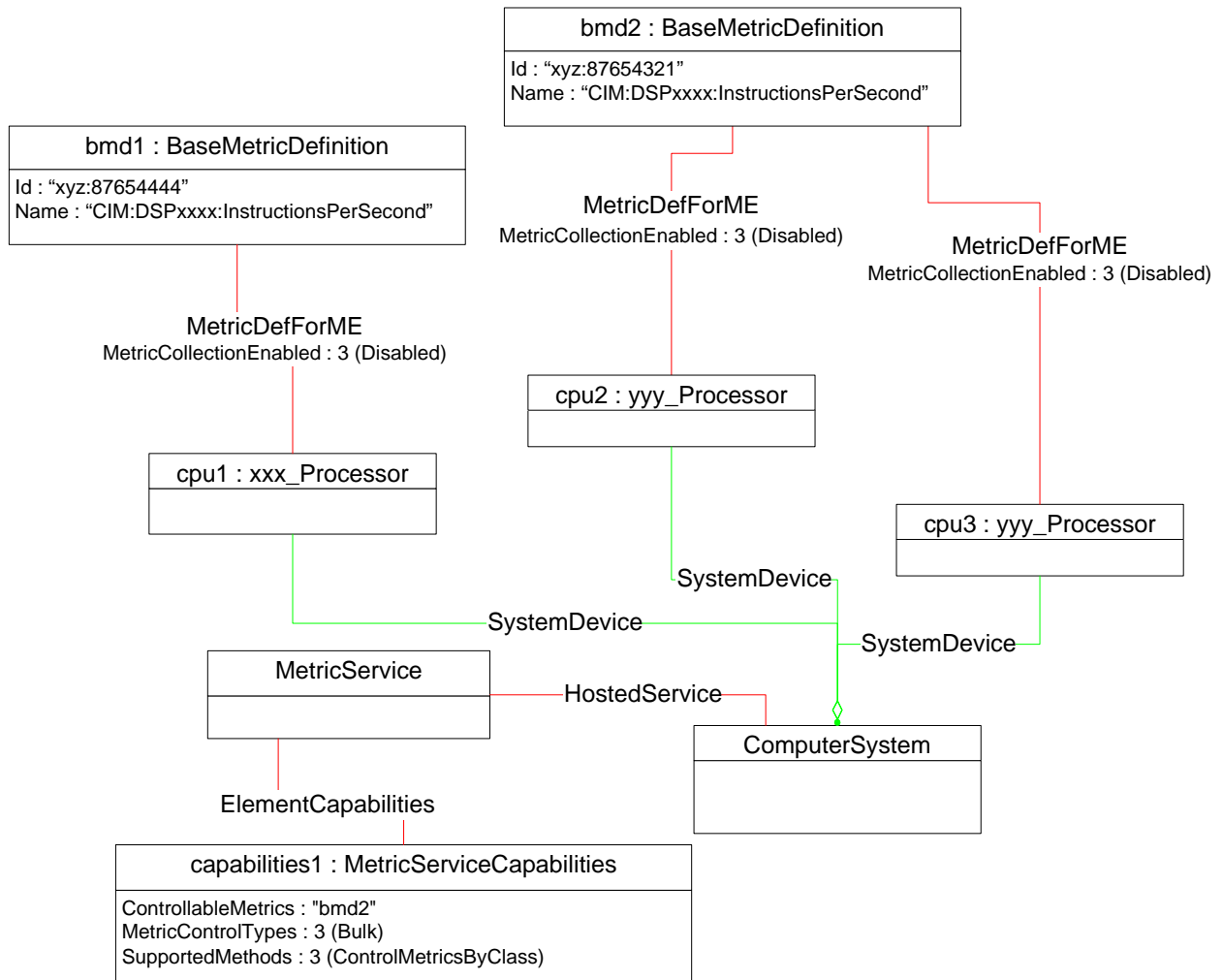


950

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

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

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



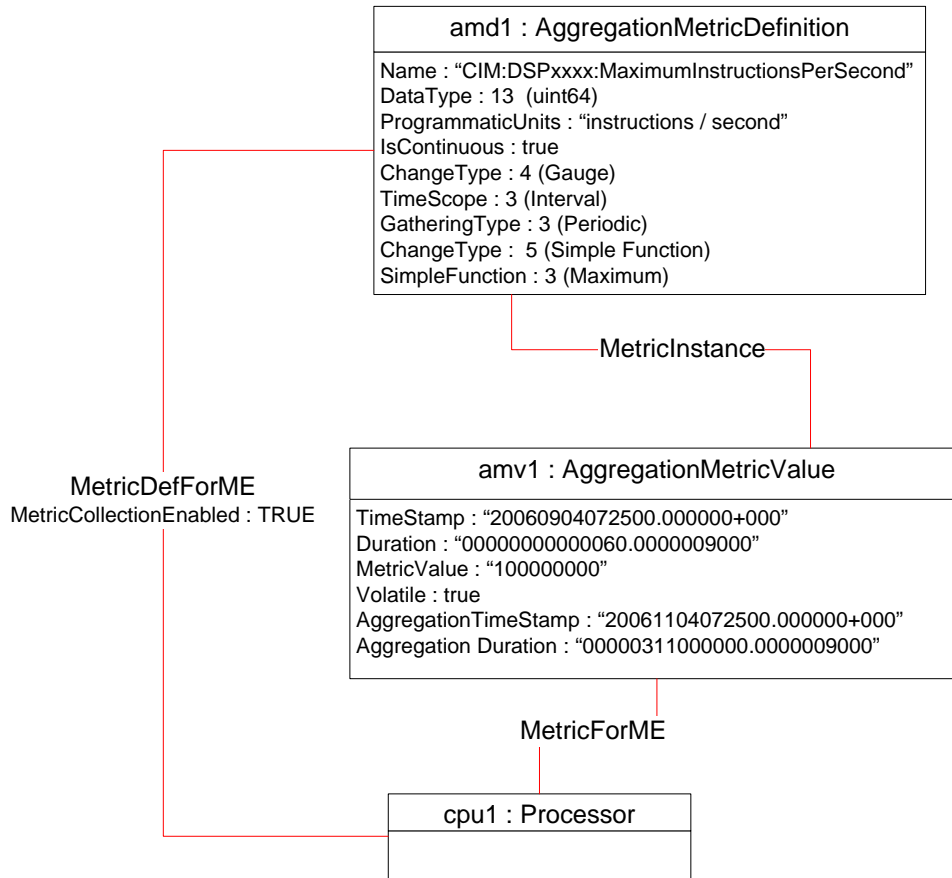
961

962

Figure 12 – Bulk Controllable Metrics by Class

963 **9.5 Aggregation Metrics**

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

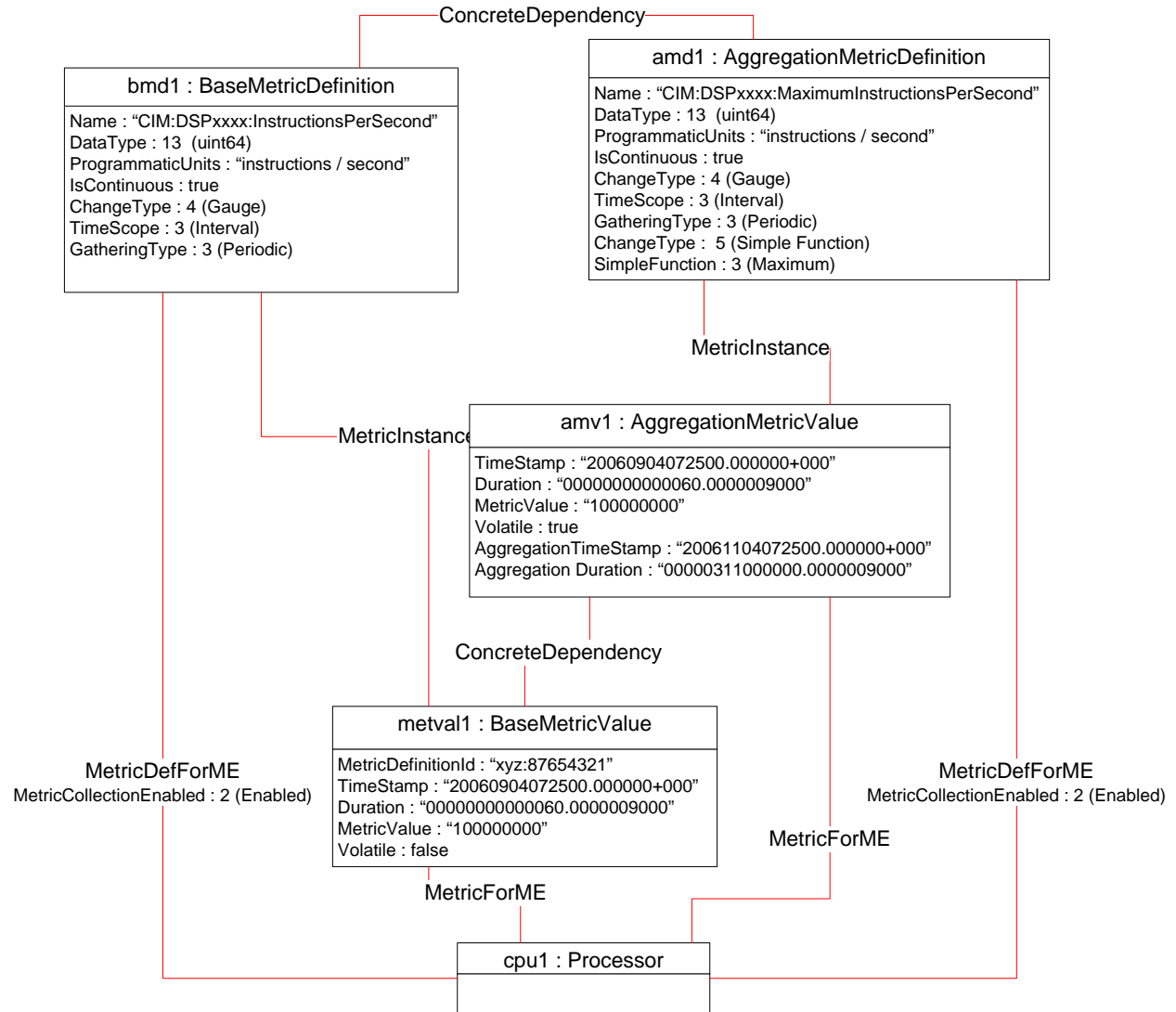


967

968

Figure 13 – Aggregation Metric without Base

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



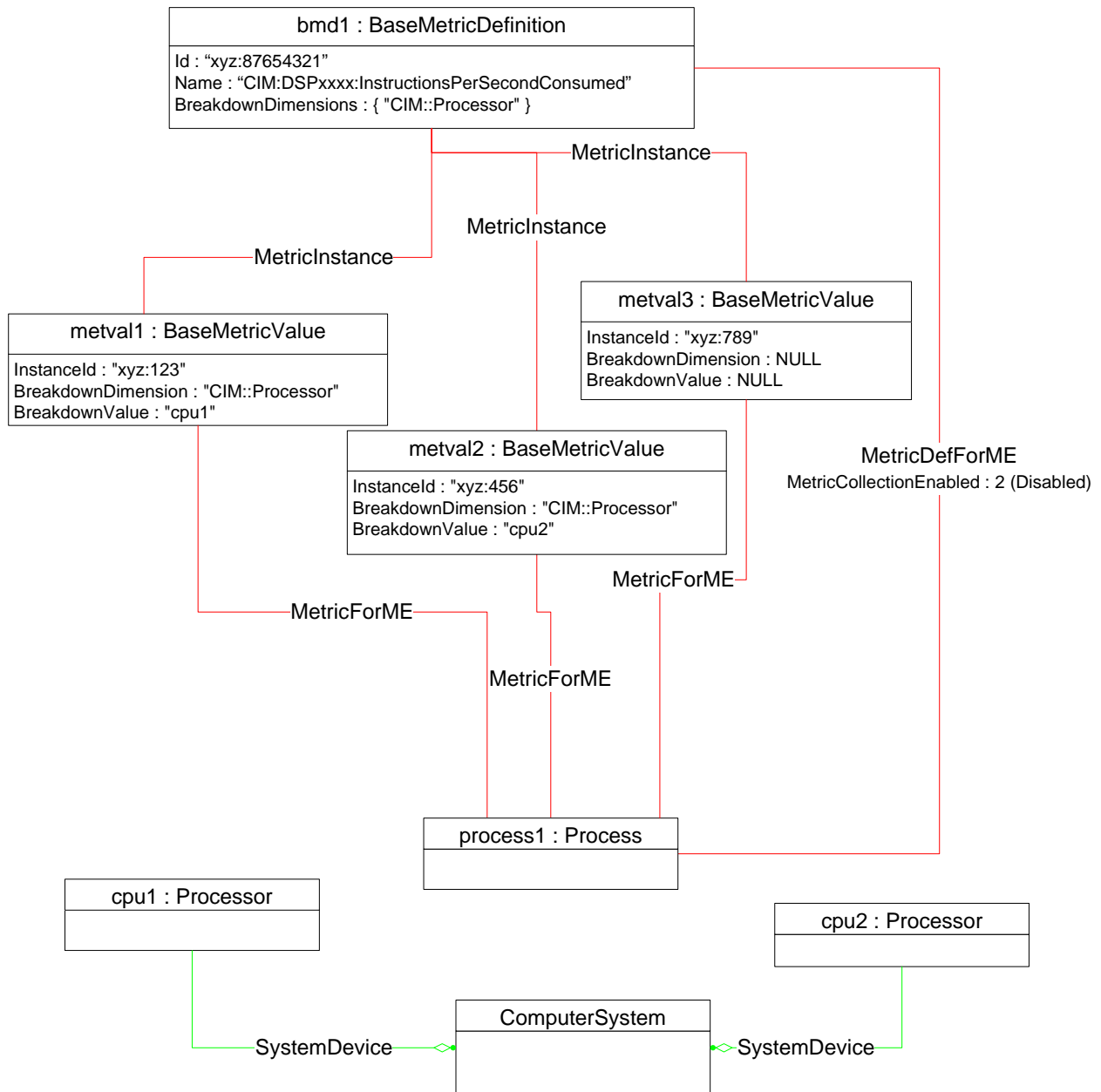
974

975

Figure 14 – Aggregation Metric with Base

976 **9.6 Metric Context**

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



982

983

Figure 15 – Metric Context

984 **9.7 Find All Metric Definitions for a Managed Element**

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

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

989 **9.8 Find the Metric Value for a Managed Element**

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

- 993 1) Find the instance of CIM_MetricDefForME that associates the CIM_BaseMetricDefinition with
994 the CIM_ManagedElement instance.
- 995 2) If the value of the MetricCollectionEnabled property of the instance of CIM_MetricDefForME
996 association found in the previous step is 2 (Enabled):
- 997 a) Find all instances of CIM_BaseMetricValue associated with the CIM_BaseMetricDefinition
998 through the CIM_MetricInstance association.
- 999 b) Find all instances of CIM_BaseMetricValue associated with the CIM_ManagedElement
1000 instance through CIM_MetricForME.
- 1001 c) Find the instance of CIM_BaseMetricValue that is the intersection of the two result sets by
1002 finding matching InstanceID property values.
- 1003 3) Otherwise, metric collection is disabled and a current data metric value is not available.

1004 **9.9 Find a Standard Metric for a Managed Element**

1005 Given a string value corresponding to the unique identifier of a standard metric, a client can find the
1006 standard metric value for an instance of CIM_ManagedElement as follows:

- 1007 1) Use the steps in 9.6 to find all metric definitions available for the instance of
1008 CIM_ManagedElement.
- 1009 2) For each instance of CIM_BaseMetricDefinition returned, determine if the Name property
1010 matches the string identifier. If there is a match, use the steps in 9.8 to find the metric value.
- 1011 3) If a matching Name property is not found, the standard metric is not supported for the instance
1012 of CIM_ManagedElement.

1013 **9.10 Retrieve a Metric Value**

1014 A client can retrieve a metric value as follows:

- 1015 1) Using the steps in 9.9, find the instance of CIM_BaseMetricValue that reports the metric.
- 1016 2) Invoke GetInstance to query the current values of properties of the CIM_BaseMetricValue
1017 instance.

1018 **9.11 Find All Metrics Available for a Managed Element within an Enumeration**
1019 **Scope**

1020 Given an instance of CIM_ManagedElement, a client can find all of the metrics available for an instance
1021 of CIM_ManagedElement as follows:

- 1022 1) Enumerate all instances of CIM_MetricService within the enumeration scope.
 - 1023 a) For each instance of CIM_MetricService, find the instance of
1024 CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities.
 - 1025 b) Query the value of the CIM_MetricServiceCapabilities.SupportedMetrics property.
 - 1026 c) If the array contains the value 4 (Show Metrics), invoke the
1027 CIM_MetricService.ShowMetrics() method providing the reference to the
1028 CIM_ManagedElement.
 - 1029 d) The list of references to CIM_BaseMetricDefinition returned as the value of the Definitions
1030 parameter identifies instances of CIM_BaseMetricDefinition that are available for the
1031 CIM_ManagedElement instance.
- 1032 2) Starting with a reference to the instance of CIM_ManagedElement, find all instances of
1033 CIM_MetricDefinition that are associated through CIM_MetricDefForME.
- 1034 3) Union the results of d) and 2).

1035 **9.12 Find All Metrics Available within an Enumeration Scope for All Instances of a**
1036 **CIM Class**

1037 Given a CIM class name, a client can find all of the metrics available within an enumeration scope for all
1038 instances of the class as follows:

- 1039 1) Enumerate all instances of CIM_MetricService within the enumeration scope.
 - 1040 a) For each instance of CIM_MetricService, find the instance of
1041 CIM_MetricServiceCapabilities associated through CIM_ElementCapabilities.
 - 1042 b) Query the value of the CIM_MetricServiceCapabilities.SupportedMetrics property.
 - 1043 c) If the array contains the value 5 (ShowMetricsByClass), invoke the
1044 CIM_MetricService.ShowMetrics() method providing the reference to the
1045 CIM_ManagedElement.
 - 1046 d) The list of references to CIM_BaseMetricDefinition returned as the value of the Definitions
1047 parameter identifies instances of CIM_BaseMetricDefinition that are available for the
1048 CIM_ManagedElement instance.
- 1049 2) Enumerate all instances of the CIM class.
 - 1050 a) For each instance of the CIM class, find all instances of CIM_BaseMetricDefinition that are
1051 associated through CIM_MetricDefForME.
- 1052 3) Form a set of instances of CIM_BaseMetricDefinition from the intersection of the instances
1053 returned by a).
- 1054 4) Union the results of d) and 3).

1055 **9.13 Determine whether a Metric Can Be Discretely Controlled for a Specific** 1056 **Managed Element**

1057 Given an instance of CIM_BaseMetricDefinition associated to an instance of CIM_ManagedElement
1058 through CIM_MetricDefForME, a client can determine whether the metric can be controlled for the
1059 managed element as follows:

- 1060 1) Starting with the instance of CIM_BaseMetricDefinition, find the instance of CIM_MetricService
1061 associated through CIM_ServiceAffectsElement.
- 1062 2) Find the instance of CIM_MetricServiceCapabilities associated through
1063 CIM_ElementCapabilities with the instance of CIM_MetricService found in 1).
- 1064 3) If the following conditions are met, the metric can be discretely controlled for the managed
1065 element:
 - 1066 a) The CIM_MetricServiceCapabilities.ControllableMetrics property contains a reference to
1067 the CIM_BaseMetricDefinition.
 - 1068 b) The CIM_MetricServiceCapabilities.MetricControlTypes property contains the value 2
1069 (Discrete) or 4 (Both) at the same array index as the reference in the previous step.
 - 1070 c) The CIM_MetricServiceCapabilities.ControllableManagedElements property contains a
1071 reference to the CIM_ManagedElement.
 - 1072 d) The CIM_MetricServiceCapabilities.ManagedElementControlTypes property contains the
1073 value 2 (Discrete) or 4 (Both) at the same array index as the reference in the previous step.
- 1074 4) Otherwise, the metric can not be discretely controlled for the managed element.

1075 **9.14 Enable a Specific Metric for a Specific Managed Element**

1076 Given an instance of CIM_BaseMetricDefinition that is associated to an instance of
1077 CIM_ManagedElement through the CIM_MetricDefForME association, a client can enable a specific
1078 metric for the managed element as follows:

- 1079 1) Use the steps in 9.13 to determine whether the metric can be controlled.
- 1080 2) Invoke the CIM_MetricService.ControlMetrics() method and specify the reference to the
1081 CIM_ManagedElement as the value of the Subject parameter, the reference to the
1082 CIM_BaseMetricDefinition as the value of the Definition parameter, and TRUE as the value of
1083 the MetricCollectionEnabled parameter.

1084 **9.15 Find All Managed Elements within an Enumeration Scope for which a Metric** 1085 **Is Currently Being Collected**

1086 Given an instance of CIM_BaseMetricDefinition, a client can find all instances of CIM_ManagedElement
1087 for which the metric is available as follows:

- 1088 1) Find the instance of CIM_MetricService associated with the CIM_BaseMetricDefinition through
1089 the CIM_ServiceAffectsElement association.
- 1090 2) Find the instance of CIM_MetricServiceCapabilities associated with the CIM_MetricService
1091 found in the previous step.
- 1092 3) Query the value of the SupportedMethods property of the instance of
1093 CIM_MetricServiceCapabilities found in the previous step to determine if it contains a value of 4
1094 (ShowMetrics):
 - 1095 a) If the SupportedMethods property contains a value of 4 (ShowMetrics):
 - 1096 – Invoke the CIM_MetricService.ShowMetrics() method, specifying the reference to the
1097 CIM_BaseMetricDefinition as the value of the Definition parameter.

- 1098 – Upon successful completion of the method, the ManagedElements parameter
- 1099 contains a list of references to CIM_ManagedElement instances for which the metric
- 1100 defined by the CIM_BaseMetricDefinition is available. The MetricCollectionEnabled
- 1101 property indicates whether the metric is currently being collected for the
- 1102 CIM_ManagedElement instance.

- 1103 4) If the SupportedMethods property does not contain the value 4, find all instances of
- 1104 CIM_MetricDefForME that reference the CIM_BaseMetricDefinition instance. For each instance
- 1105 of CIM_MetricDefForME, the Antecedent property identifies a CIM_ManagedElement for which
- 1106 the metric may be collected and the MetricCollectionEnabled property indicates whether the
- 1107 metric is currently being collected.

1108 **10 CIM Elements**

1109 Table 19 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be
 1110 implemented as described in Table 19. Sections 7 ("Implementation") and 8 ("Methods") may impose
 1111 additional requirements on these elements.

1112 **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		

1113 **10.1 CIM_AggregationMetricDefinition**

1114 CIM_AggregationMetricDefinition defines a metric that can be captured. Table 20 defines the
 1115 requirements for instances of CIM_AggregationMetricDefinition.

1116 **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

1117 **10.2 CIM_AggregationMetricDefinition (Low Watermark)**

1118 Table 21 defines the requirements for instances of CIM_AggregationMetricDefinition used to define a low
 1119 watermark metric.

1120 **Table 21 – Class: CIM_AggregationMetricDefinition (Low Watermark)**

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

1121 **10.3 CIM_AggregationMetricDefinition (High Watermark)**

1122 Table 22 defines the requirements for instances of CIM_AggregationMetricDefinition used to define high
 1123 watermark metrics.

1124 **Table 22 – Class: CIM_AggregationMetricDefinition (High Watermark)**

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

1125 **10.4 CIM_AggregationMetricValue**

1126 CIM_AggregationMetricValue conveys the actual recorded data of a metric that has been maintained.
 1127 Table 23 describes the requirements for instances of CIM_AggregationMetricValue.

1128 **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

1129 **10.5 CIM_BaseMetricDefinition**

1130 CIM_BaseMetricDefinition defines a metric that can be captured. Table 24 defines the requirements for
 1131 instances of CIM_BaseMetricDefinition.

1132 **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.

1133 **10.6 CIM_BaseMetricDefinition — Instantaneous Metric**

1134 Table 25 describes the requirements for using CIM_BaseMetricDefinition to define an Instantaneous
 1135 Metric. These constraints are in addition to those specified in 10.5.

1136 **Table 25 – Class: CIM_BaseMetricDefinition – Instantaneous Metric**

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

1137 **10.7 CIM_BaseMetricDefinition — Interval Metric**

1138 Table 26 describes the requirements for using CIM_BaseMetricDefinition to define an Interval Metric.
 1139 These constraints are in addition to those specified in 10.5.

1140 **Table 26 – Class: CIM_BaseMetricDefinition – Interval Metric**

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 3 (Interval)

1141 **10.8 CIM_BaseMetricDefinition — Startup Interval Metric**

1142 Table 27 describes the requirements for using CIM_BaseMetricDefinition to define a Startup Interval
 1143 Metric. These constraints are in addition to those specified in 10.5.

1144 **Table 27 – Class: CIM_BaseMetricDefinition – Startup Interval Metric**

Properties	Requirement	Notes
TimeScope	Mandatory	Matches 4 (Startup Interval)

1145 **10.9 CIM_BaseMetricDefinition — Summation Metric**

1146 Table 28 describes the requirements for using CIM_BaseMetricDefinition to define a Summation Metric.
 1147 These constraints are in addition to those specified in 10.5.

1148 **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)

1149 **10.10 CIM_BaseMetricDefinition — Current Data**

1150 Table 29 describes the requirements for using CIM_BaseMetricDefinition to define the metric to be used
 1151 with current data. These constraints are in addition to those specified in 10.5.

1152 **Table 29 – Class: CIM_BaseMetricDefinition – Current Data**

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

1153 **10.11 CIM_BaseMetricValue**

1154 CIM_BaseMetricValue conveys the actual recorded data of a metric. Table 30 describes the requirements
 1155 for instances of CIM_BaseMetricValue.

1156 **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

1157 **10.12 CIM_BaseMetricValue — Current Data**

1158 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 31 describes the
 1159 requirements for using CIM_BaseMetricValue to report the metric for current data. These constraints are
 1160 in addition to those specified in 10.11.

1161 **Table 31 – Class: CIM_BaseMetricValue – Current Data**

Properties	Requirement	Notes
Timestamp	Mandatory	None
Volatile	Mandatory	Matches TRUE

1162 **10.13 CIM_BaseMetricValue — Interval Metrics**

1163 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 32 describes the
 1164 requirements for using CIM_BaseMetricValue to report the metric for interval metrics. These constraints
 1165 are in addition to those specified in 10.11.

1166 **Table 32 – Class: CIM_BaseMetricValue – Interval Metrics**

Properties	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None

1167 **10.14 CIM_BaseMetricValue — Startup Interval Metrics**

1168 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 33 describes the
 1169 requirements for using CIM_BaseMetricValue to report the metric for startup interval metrics. These
 1170 constraints are in addition to those specified in 10.11.

1171 **Table 33 – Class: CIM_BaseMetricValue – Startup Interval Metrics**

Properties	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None

1172 **10.15 CIM_BaseMetricValue — Summation Metric**

1173 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 34 describes the
 1174 requirements for using CIM_BaseMetricValue to report the metric for a Summation Metric. These
 1175 constraints are in addition to those specified in 10.11.

1176 **Table 34 – Class: CIM_BaseMetricValue – Summation Metric**

Properties	Requirement	Notes
Timestamp	Mandatory	None

1177 **10.16 CIM_BaseMetricValue — Long-Term Monitoring**

1178 CIM_BaseMetricValue reports a metric defined using CIM_BaseMetricDefinition. Table 35 describes the
 1179 requirements for using CIM_BaseMetricValue to report a metric for long-term monitoring. These
 1180 constraints are in addition to those specified in 10.11.

1181 **Table 35 – Class: CIM_BaseMetricValue – Long-Term Monitoring**

Properties	Requirement	Notes
Volatile	Mandatory	Matches FALSE

1182 **10.17 CIM_ConcreteDependency (Definition)**

1183 Table 36 details the requirements for instances of CIM_ConcreteDependency.

1184 **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

1185 **10.18 CIM_ConcreteDependency (Value)**

1186 Table 37 details the requirements for instances of CIM_ConcreteDependency.

1187 **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

1188 **10.19 CIM_ElementCapabilities**1189 CIM_ElementCapabilities associates an instance of CIM_MetricServiceCapabilities with the Central
1190 Instance. Table 38 details the requirements for instances of CIM_ElementCapabilities.1191 **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

1192 **10.20 CIM_HostedService**

1193 Table 39 details the requirements for instances of CIM_HostedService.

1194 **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..*

1195 **10.21 CIM_MetricDefForME**

1196 CIM_MetricForME relates a metric to the managed element for which it was measured. Table 40 details
 1197 the requirements for instances of CIM_MetricDefForME.

1198 **Table 40 – Class: CIM_MetricDefForME**

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

1199 **10.22 CIM_MetricForME**

1200 CIM_MetricForME relates a metric to the managed element for which it was measured. Table 41 details
 1201 the requirements for instances of CIM_MetricForME.

1202 **Table 41 – Class: CIM_MetricForME**

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

1203 **10.23 CIM_MetricInstance**

1204 CIM_MetricInstance relates a CIM_BaseMetricValue to the CIM_BaseMetricDefinition that defines it.
 1205 Table 42 details the requirements for instances of CIM_MetricInstance.

1206 **Table 42 – Class: CIM_MetricInstance**

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

1207 **10.24 CIM_MetricService**

1208 Table 43 details the requirements for instances of CIM_MetricService.

1209 **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.

1210 **10.25 CIM_MetricServiceCapabilities**

1211 CIM_MetricServiceCapabilities indicates support for managing the state of the service as well as the
 1212 accounts with which the service is associated. Table 44 details the requirements for instances of
 1213 CIM_MetricServiceCapabilities.

1214 **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

1215 **10.26 CIM_RegisteredProfile**

1216 CIM_RegisteredProfile identifies the *Base Metrics Profile*. The CIM_RegisteredProfile class is defined by
 1217 the [Profile Registration Profile](#). With the exception of the mandatory values specified for the properties in
 1218 Table 45, the behavior of the CIM_RegisteredProfile instance is in accordance with the constraints
 1219 specified in the [Profile Registration Profile](#).

1220 **Table 45 – Class: CIM_RegisteredProfile**

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

1221 **10.27 CIM_ServiceAffectsElement**

1222 CIM_ServiceAffectsElement is used to associate an instance of CIM_MetricService with an instance of
 1223 CIM_BaseMetricDefinition or CIM_AggregationMetricDefinition that represents a metric that could be
 1224 controlled using the service. Table 46 contains the requirements for elements of this class.

1225 **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)

1226

ANNEX A
(Informative)**Change Log**

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

1227
1228
1229
1230
1231

1232

ANNEX B (Informative)

Guide for Common Metrics

1233
1234
1235
1236

1237 This annex provides an informative list of the combined mandatory properties for instances of
1238 CIM_BaseMetricDefinition, CIM_AggregationMetricDefinition, CIM_BaseMetricValue, and
1239 CIM_AggregationMetricValue if used to represent common metrics. Each of the data cells of the tables
1240 lists mandatory properties and their value formulations for a specific type of metric. Each table
1241 corresponds to a different type of metrics grouped by value formulation. The rows represent the different
1242 type of metrics based on the time scope that metric describes. The columns describe the different type of
1243 metrics based on the metric collection access type used.

1244 In order to determine the mandatory set of properties, match the type of metric to one of the data cells
1245 based on the metric's value formulation, time scope, and collection access type.

1246 The following conventions are used in the table:

- 1247 • BMD – the properties that follow are required on the instance of CIM_BaseMetricDefinition
- 1248 • BMV – the properties that follow are required on the instance of CIM_BaseMetricValue
- 1249 • AMD – the properties that follow are required on the instance of
1250 CIM_AggregationMetricDefinition
- 1251 • AMV – the properties that follow are required on the instance of CIM_AggregationMetricValue
- 1252 • A property name without a value specified is required, and the value is not fixed
- 1253 • A property name followed by a value assignment is required with the specified value fixed.

1254 NOTE: If there is a mismatch between the mandatory set of properties and/or the properties' value formulation
1255 indicated by the tables in this annex and the requirements detailed in 7 and 10, the requirements mandated in 7 and
1256 10 take precedence.

1257 **B.1 Simple Metric**

1258 Table B-1 describes the mandatory properties for simple metric as described in 6.3.1 according to the
1259 appropriate metric access type and time scope.

1260

Table B-1 – Simple Metric

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

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

1261 **B.2 Summation Metric**

1262 Table B-2 describes the mandatory properties for summation metric as described in 6.3.2 according to
 1263 the appropriate metric access type and time scope.

1264 **Table B-2 – Summation Metric**

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

	Current Data	Current Data - Online Monitoring	Current Data - Snapshot Monitoring	Long-Term Monitoring	Event-Based Monitoring
Interval Metrics	<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>Volatile = TRUE</p> <p>Volatile</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

1265 **B.3 Aggregation Metric**

1266 Table B-3 describes the mandatory properties for aggregation metric as described in section 6.3.3
 1267 according to the appropriate metric access type and time scope.

1268 **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	AggregationTimeSta mp	AggregationTimeSta mp
	Timestamp	Timestamp	Timestamp	AggregationDuration	AggregationDuration
AggregationTimeSta mp	AggregationTimeSta mp	AggregationTimeSta mp			
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 ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id ITimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType Id TimeScope = 3 (Interval) ProgrammaticUnits Name SimpleFunction</p>
	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamps AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId Volatile = TRUE Volatile InstanceID Duration Timestamp AggregationTimeStamps AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStamps AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeStamps AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeStamps AggregationDuration</p>
Startup Metrics	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name 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
	AggregationTimeStamps	AggregationTimeStamps	AggregationTimeStamps	AggregationTimeStamps	AggregationTimeStamps
	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration

1269 **B.4 Aggregation Metric — Low Watermark**

1270 Table B-4 describes the mandatory properties for low watermark as a type of an aggregation metric as
 1271 described in section 6.3.3.1 according to the appropriate metric access type and time scope.

1272 **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 AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = FALSE InstanceID AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile InstanceID AggregationTimeSta mp 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 AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId Volatile = TRUE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeSta mp 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) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 3 (Periodic) or 2 (OnChange) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	ChangeType = 5 (Simple Function) DataType ElementName GatheringType = 4 (OnRequest) Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)	ChangeType = 5 (Simple Function) DataType ElementName GatheringType Id TimeScope = 4 (StartupInterval) ProgrammaticUnits Name SimpleFunction = 2 (Minimum)
	AMV	AMV	AMV	AMV	AMV
	MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeStam mp AggregationDuration	MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration	MetricDefinitionId MetricValue Volatile = TRUE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration	MetricDefinitionId MetricValue Volatile = FALSE InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration	MetricDefinitionId MetricValue Volatile InstanceID Duration Timestamp AggregationTimeSta mp AggregationDuration

1273 **B.5 Aggregation Metric — High Watermark**

1274 Table B-5 describes the mandatory properties for high watermark as a type of an aggregation metric as
 1275 described in section 6.3.3.1 according to the appropriate metric access type and time scope.

1276 **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	AggregationTimeSta mp	AggregationTimeSta mp	
Timestamp	Timestamp	Timestamp	AggregationDuration	AggregationDuration	
AggregationTimeSta mp	AggregationTimeSta mp	AggregationTimeSta mp			
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 ElementName</p> <p>GatheringType Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType = 3 (Periodic) or 2 (OnChange)</p> <p>Id ITimeScope = 3 (Interval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType = 4 (OnRequest)</p> <p>Id TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType Id</p> <p>TimeScope = 3 (Interval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>
	<p>AMV</p> <p>MetricDefinitionId MetricValue</p> <p>Volatile = TRUE</p> <p>InstanceID Duration</p> <p>Timestamp AggregationTimeStam p</p> <p>AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId Volatile = TRUE</p> <p>Volatile</p> <p>InstanceID Duration</p> <p>Timestamp AggregationTimeSta mp</p> <p>AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue</p> <p>Volatile = TRUE</p> <p>InstanceID Duration</p> <p>Timestamp AggregationTimeSta mp</p> <p>AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue</p> <p>Volatile = FALSE</p> <p>InstanceID Duration</p> <p>Timestamp AggregationTimeSta mp</p> <p>AggregationDuration</p>	<p>AMV</p> <p>MetricDefinitionId MetricValue</p> <p>Volatile</p> <p>InstanceID Duration</p> <p>Timestamp AggregationTimeSta mp</p> <p>AggregationDuration</p>
Startup Metrics	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType = 3 (Periodic) or 2 (OnChange)</p> <p>Id TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType = 4 (OnRequest)</p> <p>Id TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits Name</p> <p>SimpleFunction = 3 (Maximum)</p>	<p>AMD</p> <p>ChangeType = 5 (Simple Function)</p> <p>DataType ElementName</p> <p>GatheringType Id</p> <p>TimeScope = 4 (StartupInterval)</p> <p>ProgrammaticUnits 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
	AggregationTimeStamps	AggregationTimeStamps	AggregationTimeStamps	AggregationTimeStamps	AggregationTimeStamps
	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration	AggregationDuration

1277

1278

Bibliography

1279

1280 DMTF DSP1073, *Capacity Metrics Profile 1.0* (under development)

1281