2

6

WBEM Glossary

10

Copyright notice

14

CONTENTS

15

Foreword

This document was prepared by the DMTF Architecture Working Group

16 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about the DMTF, see http://www.dmtf.org.

17
Acknowledgements

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

19 • Jim Davis, WS (Editor)

20 • Andreas Maier, IBM

21 • George Ericson, EMC

22 • Karl Schopmeyer, Inova

23
Document conventions

24
Typographical conventions

25 The following typographical conventions are used in this document:

26 • Document titles are marked in *italics*.

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

28 • Terms include a link to the term definition in the "Terms and definitions" clause, enabling easy navigation to the term definition.

29

WBEM Glossary

30

1 Scope

31 The WBEM Glossary is a normative glossary that defines terms and abbreviations for use by other documents.

32 ## 1.1 Usage

33 Other documents that utilize this glossary are termed "using documents". Using documents shall include this glossary document into their "Normative references" clause.

34 If one or more terms defined in this glossary are used, the using document shall place a sentence similar to the following into the introduction subclause of their "Terms and definitions" clause:

The terms defined in DSP0198 apply to this document.

35 If one or more abbreviations defined in this glossary are used, the using document shall place a sentence similar to the following into the introduction subclause of their "Symbols and abbreviations" clause:

The abbreviations defined in DSP0198 apply to this document.

36 For each term or abbreviation that is used from this glossary, using documents should apply one of the following approaches:

37 • Simply using the term or abbreviation without redefining it. This is the recommended approach.

39 • Repeating the definition of the term or abbreviation consistent with the definition in this glossary, either with the full text or with an abridged text, referencing this glossary as the authoritative source. This approach can be used if the term is so important for the using document that a local definition seems appropriate. For example (using abridged definitions):

40 **association**
a relationship between classes, as defined in DSP0198

41 **aggregation**
a strong form of association; for a full definition see DSP0198

43 • Defining a term or abbreviation different from the definition in this glossary. This approach should only be used if there are good reasons for the differing definition. If this approach is used, the using document needs to clarify whether its definition overrides or amends the definition in this glossary. For example:

44 **composition**

45 the process of creating a new piece of music.

46 This definition overrides the definition in DSP0198.

47 **ECMA**

48 European Carton Makers Association; see http://www.ecma.org

49 This definition amends the definition in DSP0198.

50

1.2 Referencing terms and abbreviations

51 This glossary supports external references to its terms and abbreviations.

52 In the PDF version of this glossary, each term and each abbreviation is available as a PDF *named destination*. This enables hyperlinks of the form:

53 `<url-to-pdf>#nameddest=Term_<term-name>`
`<url-to-pdf>#nameddest=Sym_<symbol-name>`

54 where `<term-name>` and `<symbol-name>` are the normalized terms and abbreviations, respectively. The normalization is case-preserving and changes any characters other than alphanumeric characters into underscore.

55 Examples:

56 • Term "CIM object" becomes named destination `Term_CIM_object`, and a URL to its definition in version 1.0 of this glossary is:
http://www.dmtf.org/.../DSP0198_1.0.pdf#nameddest=Term_CIM_object

57 • Abbreviation "DMTF" becomes named destination `Sym_DMTF`, and a URL to its definition in version 1.0 of this glossary is:
http://www.dmtf.org/.../DSP0198_1.0.pdf#nameddest=Sym_DMTF

58 Documents authored in a word processing format that supports hyperlinks can simply set these URLs in hyperlinks.

59 Documents authored in DMTF MRP XML format (machine readable profiles) can use the `docRefName` attribute of `<mrp:TermLink>` and `<mrp:SymbolLink>` elements to link to the MRP XML version of this glossary, as follows (assuming the profile specifies a normative reference to this glossary using the name `"Ref_DMTF_DSP0198"`:

60 • Link to term "CIM object" in MRP XML:
```
<mrp:TermLink docRefName="Ref_DMTF_DSP0198"
    refName="Term_CIM_object">CIM object</mrp:TermLink>
```

61 • Link to abbreviation "DMTF" in MRP XML:
```
<mrp:SymbolLink docRefName="Ref_DMTF_DSP0198"
    refName="Sym_DMTF">DMTF</mrp:SymbolLink>
```

62

2 Normative references

63 The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

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

65

3 Terms and definitions

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

67

3.1 General

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

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

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

71   The following additional terms are defined in this document.

72   **3.2**

73   **abstract class**
a class that is abstract and serves only as a base for new classes. It is not possible to create instances of such classes.

74   **3.3**

75   **abstract class adaptation**
a class adaptation that is not directly implemented and whose requirements are propagated into derived class adaptations

76   **3.4**

77   **abstract profile**
a special kind of profile specifying common elements and behavior as a base for derived profiles

78   **3.5**

79   **adaptation**
a synonym for class adaptation

80   **3.6**

81   **aggregation**

82   a strong form of association that expresses a whole-part relationship between each instance on the aggregating end and the instances on the other ends, where the instances on the other ends can exist independently from the aggregating instance.

83   For example, the containment relationship between a physical server and its physical components can be considered an aggregation, since the physical components can exist if the server is dismantled.

84   A stronger form of aggregation is a composition.

85   **3.7**

86   **arity**
the number of references exposed by an association class

87   **3.8**

88   **association**
short for CIM association

89

3.9

90 **association end**

a synonym for the reference defined in an association

91 **3.10**

92 **autonomous profile**

a profile that addresses an autonomous and self-contained management domain

93 **3.11**

94 **base adaptation**

a class adaptation that is used as the base for another class adaptation

95 **3.12**

96 **base profile**

a profile that is used as the base for another profile

97 **3.13**

98 **cardinality**

the number of instances in a set

99 **3.14**

100 **CIM-XML**

101 a WBEM protocol that uses an XML encoding over HTTP.

102 CIM-XML is defined in the following specifications;

103 • [DSP0200 (CIM Operations over HTTP)](#)

104 • [DSP0201 (Representation of CIM in XML)](#)

105 • [DSP0203 (CIM-XML DTD)](#)

106 • [DSP8044 (CIM-XML XSD)](#)

108 **3.15**

109 **CIM-RS**

110 a WBEM protocol that uses REST.

111 CIM-RS is defined in the following specifications;

112 • [DSP0210 (CIM-RS Protocol)](#)

113 • [DSP0211 (CIM-RS Payload Representation in JSON)](#)

115 **3.16**

116 **CIM association**

117 a special kind of class that expresses the relationship between two or more other classes.

118 Because associations are classes and therefore may have instances, the term "association" is often used together with a qualification, as "association class", or "association instance".

119 The relationship is established by two or more references defined in the association that are typed to a class the referenced instances are of.

120 For example, an association ACME_SystemDevice may relate the classes ACME_System and ACME_Device by defining references to those classes. A CIM association is a UML association class. Each has the aspects of both a UML association and a UML class, which may expose ordinary properties

and methods and may be part of a class inheritance hierarchy. The references belonging to a CIM association belong to it and are also exposed as part of the association and not as parts of the associated classes. The term "association class" is sometimes used instead of the term "association" when the class aspects of the element are being emphasized.

121   Aggregations and compositions are special kinds of associations.

122   **3.17**

123   **CIM class**

124   a common type for a set of instances that share the same properties, methods, constraints, and semantics.

125   A class is defined in a schema and models an aspect of a managed object. For example, a class named "ACME_Modem" (defined in the schema named "ACME") might model some aspect of modems and might define a property named "ActualSpeed" to indicate the actual modem speed.

126   Special kinds of classes are ordinary classes, association classes and indication classes.

127   In a CIM server, classes are addressable objects. The term "class object" (i.e., object of class type) is sometimes used to emphasize that. The address of class objects is termed "class path".

128   In a schema, classes are special kinds of schema elements.

129   **3.18**

130   **CIM client**

131   a role participating in a CIM protocol that is responsible for originating CIM operations for processing by a CIM server.

132   This definition does not imply any particular implementation architecture or scope, such as a client library component or an entire management application.

133   **3.19**

134   **CIM indication**

135   This term has two meanings:

136   • a special kind of class that expresses the notification about an event that occurred.

137   • an interaction within a CIM protocol that is originated on a CIM server and processed by a CIM listener.

138   Because indications are classes and therefore may have instances, and because of the additional meaning as protocol interaction, the term "indication" is often used together with a qualification, as "indication class", "indication instance", or "indication interaction".

139   Indications are raised based on a trigger that defines the condition under which an event causes an indication to be raised. Events may be related to objects accessible in a CIM server, such as the creation, modification, deletion of or access to an object, or execution of a method on the object. Events may also be related to managed objects, such as alerts or errors. For example, an indication ACME_AlertIndication may express the notification about an alert event.

140   In a CIM server, indication instances are not addressable objects; they exist only as embedded instances in the protocol interaction that delivers the indication.

141   **3.20**

142   **CIM instance**

143   a specific realization of a class, that has values (including possible Null) for the properties exposed by that class.

144

Embedded instances are also instances.

145 In a CIM server, instances are addressable objects. The term "instance object" (i.e., object of instance type) is sometimes used to emphasize that. The address of instance objects is termed "instance path".

146 In a schema, instances are special kinds of schema elements.

147 **3.21**

148 **CIM listener**

149 a role participating in a CIM protocol that is responsible for processing indications originated by a CIM server.

150 This definition does not imply any particular implementation architecture or scope, such as a standalone demon component or an entire management application.

151 **3.22**

152 **CIM method**

153 a behavioral feature of a class.

154 Methods can be invoked to produce the associated behavior.

155 In a schema, methods are special kinds of schema elements.

156 **3.23**

157 **CIM namespace**

158 a namespace in a CIM server for classes, instances and qualifier types.

159 CIM namespaces are flat; a namespace does not contain other namespaces.

160 **3.24**

161 **CIM object**

162 a class, instance, qualifier type, or namespace that is accessible through a CIM server.

163 An object may be addressable, i.e., have an object path. Embedded objects are also objects, but they are not independently addressable; they are accessible indirectly through their embedding properties in other objects, or embedding parameters or return values in method invocations. Indication instances are also objects, but they are not independently addressable; they only exist in the protocol message in which they are being delivered.

164

165 **DEPRECATED:** The term "object" has historically be used to mean just "class or instance". This use of the term "object" is deprecated. If a restriction of the term "object" to mean just "class or instance" is intended, this is now stated explicitly.

166

167 **3.25**

168 **CIM operation**
an interaction within a CIM protocol that is originated by a CIM client and processed by a CIM server

169 **3.26**

170 **CIM parameter**

171 a named and typed argument passed in and out of methods

172 The return value of a method is not considered a parameter; instead it is considered part of the method.

173

In a schema, parameters are special kinds of schema elements.

174   **3.27**

175   **CIM property**

176   a named and typed structural feature of a class.

177   Name, data type, default value and other information about the property are defined in a class. Properties have values that are available in the instances of a class. The values of its properties may be used to characterize an instance.

178   For example, a class ACME_Device may define a string typed property named "Status". In an instance of class ACME_Device, the Status property may have a value "on".

179   Special kinds of properties are ordinary properties and references.

180   In a schema, properties are special kinds of schema elements.

181   **3.28**

182   **CIM protocol**

183   a protocol that is used between CIM client, CIM server and CIM listener.

184   This definition does not imply any particular communication protocol stack, or even that the protocol performs a remote communication.

185   **3.29**

186   **CIM qualifier**

187   a named value used to characterize schema elements.

188   Qualifier values may change the behavior or semantics of the qualified schema element. Qualifiers can be regarded as metadata that is attached to the schema elements. The scope of a qualifier determines on which kinds of schema elements a specific qualifier can be specified.

189   For example, if property ACME_Modem.Speed has the Key qualifier specified with a value of True, this characterizes the property as a key property for the class.

190   In a schema, qualifiers are special kinds of schema elements.

191   **3.30**

192   **CIM qualifier type**

193   a common type for a set of qualifiers.

194   In a CIM server, qualifier types are addressable objects. The address of qualifier type objects is termed "qualifier type path".

195   In a schema, qualifier types are special kinds of schema elements.

196   **3.31**

197   **CIM reference**

198   an association end.

199   References are special kinds of properties in an association that reference an instance. The value of a reference is an instance path. The type of a reference is a class the referenced instance is of. The referenced instance may be of a subclass of the class specified as the type of the reference.

200

3.32

201 **CIM schema**

202 a set of classes that have the same schema name.

203 For example, DMTF publishes two CIM schemas, with schema names "CIM" and "PRS". Class CIM_System has a schema name "CIM" and is thus part of the schema named "CIM".

204 **3.33**

205 **CIM Schema**

206 the schema named "CIM" that is published by DMTF.

207 The CIM Schema defines an ontology for management. The schema named "PRS" that is also published by DMTF is a separate schema.

208 **3.34**

209 **CIM server**

210 a role participating in a CIM protocol that is responsible for processing CIM operations originated by a CIM client and for originating CIM indications for processing by a CIM listener.

211 This definition does not imply any particular implementation architecture, such as a separation into an object manager and provider components.

212 **3.35**

213 **class**
a synonym for CIM class

214 **3.36**

215 **class adaptation**

216 a named element in a profile that defines requirements and constraints on the usage of a class by that profile.

217 A class adaptation may be based on other class adaptations.

218 **3.37**

219 **class declaration**

220 a representation of a class.

221 For example, there might be CIM operations for retrieving and modifying class objects that are accessible through a CIM server. The retrieval operation might return the class declaration, and the modifying operation might take the class declaration as input.

222 **3.38**

223 **class path**
a special kind of object path, addressing a class object that is accessible through a CIM server

224 **3.39**

225 **class origin**
the class defining a particular property or method

226 **3.40**

227 **Common Information Model**

228 CIM (Common Information Model) is:

229   1.   the name of the architecture and meta-model used to define CIM schemas

230   2.   the name of the CIM schema published by the DMTF (that is, the CIM Schema)

231   **3.41**

232   **common model**

233   the subset of the CIM Schema that is specific to particular management domains.

234   It is derived from the core model and is actually a collection of models, including (but not limited to) the System model, the Application model, the Network model, and the Device model.

235   **3.42**

236   **component profile**
        a profile that addresses a subset of a management domain

237   **3.43**

238   **composition**

239   a strong form of aggregation that expresses a whole-part relationship between each instance on the aggregating end and the instances on the other ends, where the instances on the other ends cannot exist independently from the aggregating instance.

240   For example, the containment relationship between a running operating system and its logical devices can be considered a composition, because the logical devices cannot exist if the operating system does not exist.

241   **3.44**

242   **concrete profile**
        any profile that is not an abstract profile

243   **3.45**

244   **core model**
        the subset of the CIM Schema that is not specific to any particular management domain. The core model establishes a basis for derived models such as the common model or extension schemas.

245   **3.46**

246   **creation class**

247   the most derived class a particular instance is of.

248   The creation class of an instance can also be considered the factory of the instance (although in CIM, instances may come into existence through other means than issuing an instance creation operation).

249   **3.47**

250   **deprecated**

251   a keyword indicating that a specification element or specification-defined behavior is outdated and may be removed in a future major version of the specification.

252   Deprecation of elements or behaviors does not save implementations from providing support for the deprecated element or behavior as required by the specification.

253   **3.48**

254   **derived profile**
        a profile that is based on a referenced profile

255

3.49

256 **element**

a synonym for schema element

257 **3.50**

258 **embedded class**

a class declaration that is embedded in the value of a property, parameter, or method return value

259 **3.51**

260 **embedded instance**

an instance declaration that is embedded in the value of a property, parameter, or method return value

261 **3.52**

262 **embedded object**

an embedded class or embedded instance

263 **3.53**

264 **experimental**

a keyword indicating that a specification element or specification-defined behavior is not yet finalized and may be changed or removed at any time

265 **3.54**

266 **extension schema**

a schema that is not owned by the DMTF and whose classes are derived from the classes in the CIM Schema

267 **3.55**

268 **feature**

a synonym for profile feature

269 **3.56**

270 **filter query**

an expression in a filter query language that can be applied to a set of instances and that specifies a subset of these instances as its result

271 **3.57**

272 **flavor**

273 an attribute on a qualifier type that specifies the rules for propagation, overriding and translatability of the corresponding qualifiers.

274 For example, if a qualifier type named "Key" specifies the flavors ToSubclass and DisableOverride, the value of a specified "Key" qualifier value gets propagated to subclasses and these subclasses cannot override it.

275 **3.58**

276 **generic operation**

277 an operation defined in [DSP0223 (Generic Operations)](#).

278 Generic operations provide a protocol-neutral definition of client operations and their semantics. For most WBEM protocols, mappings have been defined between the protocol-specific operations and the generic operations.

279

3.59

280 **indication**

a synonym for CIM indication

281 **3.60**

282 **inheritance**

283 a relationship between a more general class and a more specific class.

284 An instance of the specific class is also an instance of the general class (see polymorphism). The specific class inherits the properties and methods of the general class. In an inheritance relationship, the specific class is termed "subclass" and the general class is termed "superclass".

285 For example, if a class ACME_Modem is a subclass of a class ACME_Device, any ACME_Modem instance is also an ACME_Device instance.

286 **3.61**

287 **instance**

288 a synonym for CIM instance

289 **3.62**

290 **instance declaration**

291 a representation of an instance.

292 For example, there might be CIM operations for retrieving and modifying instance objects that are accessible through a CIM server. The retrieval operation might return the instance declaration, and the modifying operation might take the instance declaration as input.

293 **3.63**

294 **instance path**

a special kind of object path, addressing an instance object that is accessible through a CIM server

295 **3.64**

296 **key**

297 This term has two meanings:

299   • the composite key of an instance (that is, the name/value pairs of its key properties)

300   The key of an instance is part of its model path.

302   • a shorthand for key property.

303 **3.65**

304 **key property**

305 a property that is used (possibly along with other key properties) to uniquely address an instance within the set of all instances of a creation class in a namespace.

306 The key properties of a class are indicated by the Key qualifier.

307 **3.66**

308 **managed environment**

309 a concrete occurrence of a management domain.

310 A managed environment is composed of managed objects.

311

3.67

312 **managed object**

313 a physical entity, a service, or other kind of resource that exists independently of its use in management.

314 A set of managed objects composes a managed environment.

315 **3.68**

316 **management domain**

an area of work or a field of activity with common management requirements, common terminology, and related management functionality

317 **3.69**

318 **management profile**

319 a management interface between a WBEM server and a WBEM client and optionally a WBEM listener that supports a particular model.

320 A profile defines a model and its behavior in the context of a management domain. Model and behavior are defined by selecting, specializing, and sometimes constraining elements from a schema and the set of operations (including indication delivery operations) for a particular purpose. A profile defines use cases on the model that illustrate client-visible behavior.

321 **3.70**

322 **message registry**

323 a published registry of message definitions, formatted as defined in DSP0228 (Message Registry XML Schema).

324 These messages can be referenced by profiles for extended error handling, or for indication delivery.

325 **3.71**

326 **metric registry**

327 a published registry of metric definitions and optionally statistics definitions, formatted as defined in DSP8020 (Metric Registry XML Schema)

328 These metric and statistic definitions can be referenced by profiles for implementing them.

329 **3.72**

330 **method**

a synonym for CIM method

331 **3.73**

332 **method signature**

the set of names and types of the parameters of a method, and the type of its return value.

333 **3.74**

334 **model**

335 a set of classes that model a specific management domain.

336 A schema could contain multiple models (that is the case in the CIM Schema), but a particular management domain could also be modeled using multiple schemas, in which case a model would consist of multiple schemas.

337

3.75

338   **model path**

the component of an object path that identifies the object within the namespace

339   **3.76**

340   **multiplicity**

341   the multiplicity of an association end is the allowable range for the number of instances that may be associated to each instance referenced by each of the other ends of the association.

342   The multiplicity is defined on a reference using the Min and Max qualifiers.

344   **3.77**

345   **namespace**

a synonym for CIM namespace

346   **3.78**

347   **namespace path**

a special kind of object path addressing a namespace object that is accessible through a CIM server

348   **3.79**

349   **object**

a synonym for CIM object

350   **3.80**

351   **object path**

352   the address of an object that is accessible through a CIM server.

353   An object path consists of a namespace path (addressing the namespace) and optionally a model path (identifying the object within the namespace).

354   **3.81**

355   **ordinary class**

a class that is neither an association class nor an indication class

356   **3.82**

357   **ordinary property**

a property that is not a reference

358   **3.83**

359   **override**

360   a relationship between like-named schema elements in an inheritance hierarchy, where the overriding element in a subclass redefines the overridden element in a superclass.

361   The purpose of an override relationship is to refine the definition of an element in a subclass. For example, a class named "ACME_Device" may define a string typed property named "Status" that may have the values "powersave", "on", or "off". A class named "ACME_Modem", subclass of ACME_Device, may override the Status property to have only the values "on" or "off", but not "powersave".

362   **3.84**

363   **parameter**

a synonym for CIM parameter

364

3.85

365 **polymorphism**

366 the ability of an instance to be of its creation class and all of its superclasses.

367 CIM operations can exhibit polymorphic behavior. For example, a CIM operation might enumerate all instances of class ACME_Device. If the instances returned can include instances of subclasses of ACME_Device, then that CIM operation exhibits polymorphic behavior.

368 **3.86**

369 **profile**

a synonym for management profile

370 **3.87**

371 **profile feature**

a profile element that groups the decisions for the implementation of one or more profile elements into a single decision

372 **3.88**

373 **property**

a synonym for CIM property

374 **3.89**

375 **qualified element**

376 a schema element that has or can have a qualifier specified in its declaration.

377 Sometimes, the qualifier in question is known from the context. For example, if a qualifier named "Counter" can be specified on properties, methods and parameters, the description of the "Counter" qualifier would refer to the element on which it can be specified, as its qualified element.

378 **3.90**

379 **qualifier**

a synonym for CIM qualifier

380 **3.91**

381 **qualifier type**

a synonym for CIM qualifier type

382 **3.92**

383 **qualifier type declaration**

384 a representation of a qualifier type.

385 For example, there might be CIM operations for retrieving and modifying qualifier type objects that are accessible through a CIM server. The retrieval operation might return the qualifier type declaration, and the modifying operation might take the qualifier type declaration as input.

386 **3.93**

387 **qualifier type path**

a special kind of object path addressing a qualifier type that is accessible through a CIM server

388

3.94

389 **qualifier value**

the value of a qualifier in a general sense, without implying whether it is the specified value, the effective value, or the default value

390 **3.95**

391 **reference**

a synonym for CIM reference

392 **3.96**

393 **referenced profile**

a profile that is listed in the profile references table of another or the same profile

394 **3.97**

395 **referencing profile**

a profile that lists the same or another profile in its profile references table

396 **3.98**

397 **registered profile**

a profile to which an implementation advertises conformance

398 **3.99**

399 **schema**

a synonym for CIM schema

400 **3.100**

401 **schema element**

402 a specific class (including associations and indications), property (including references), method, parameter, qualifier, qualifier type, or instance.

403 For example, a class ACME_C1 or a property P1 are schema elements.

404 **3.101**

405 **scope**

406 an attribute of a qualifier type that indicates the kinds of schema elements on which the corresponding qualifier can be specified.

407 For example, if a qualifier type named "Abstract" specifies a scope of Class, Association and Indication, the "Abstract" qualifier can be specified only on ordinary classes, association classes, and indication classes.

408 **3.102**

409 **subclass**

the more specific class in an inheritance relationship

410 **3.103**

411 **superclass**

the more general class in an inheritance relationship

412

3.104

413 **Web-Based Enterprise Management**

414 a set of specifications published by DMTF that define how CIM-modeled resources can be discovered, accessed and manipulated.

415 **3.105**

416 **WBEM client**

417 a CIM client that supports a WBEM protocol.

418 A WBEM client originates WBEM operations for processing by a WBEM server. This definition does not imply any particular implementation architecture or scope, such as a client library component or an entire management application.

419 **3.106**

420 **WBEM indication**

an interaction within a WBEM protocol that is originated on a WBEM server and processed by a WBEM listener

421 **3.107**

422 **WBEM listener**

423 a CIM listener that supports a WBEM protocol.

424 A WBEM listener processes WBEM indications originated by a WBEM server. This definition does not imply any particular implementation architecture or scope, such as a standalone demon component or an entire management application.

425 **3.108**

426 **WBEM operation**

an interaction within a WBEM protocol that is originated by a WBEM client and processed by a WBEM server

427 **3.109**

428 **WBEM protocol**

429 a communications protocol between WBEM client, WBEM server and WBEM listener.

430 A WBEM protocol defines how the WBEM operations and WBEM indications work, on top of an underlying communications protocol layer (for example, HTTP, SOAP, or TCP).

431 **3.110**

432 **WBEM server**

433 a CIM server that supports a WBEM protocol.

434 A WBEM server processes WBEM operations originated by a WBEM client, and originates WBEM indications for processing by a WBEM listener. This definition does not imply any particular implementation architecture, such as a separation into an object manager and provider components.

435 **3.111**

436 **WS-Management**

437 a WBEM protocol that uses SOAP.

438 WS-Management is defined in the following specifications:

439 • [DSP0226 (Web Services for Management)](#)

440

4   Symbols and abbreviated terms

The following additional abbreviations are defined in this document.

4.1
ABNF
Augmented Backus-Naur Form, defined by the IETF

4.2
API
Application Programming Interface

4.3
CIM
Common Information Model

4.4
CIM-SPL
CIM Simplified Policy Language, as defined in [DSP0231](#)

4.5
CQL
CIM Query Language, as defined in [DSP0202](#)

4.6
CQLT
CIM Query Template Language, as defined in [DSP0202](#)

4.7
DMTF
Distributed Management Task Force; see http://wwEEEET276.0088 341.6730 Tf(Dis9M5Brce) TjETBT72.0000 2 Td(http:0

4.12

468 **IETF**

Internet Engineering Task Force; see http://www.ietf.org

469 **4.13**

470 **JSON**

JavaScript Object Notation, defined by ECMA

471 **4.14**

472 **MIB**

Management Information Base, defined by the IETF

473 **4.15**

474 **MOF**

Managed Object Format, as defined in DSP0004

475 **4.16**

476 **MRP**

Machine Readable Profile, as defined in DSP8028

477 **4.17**

478 **OCL**

Object Constraint Language, defined by the OMG

479 **4.18**

480 **OMG**

Object Management Group; see http://www.omg.org

481 **4.19**

482 **PUG**

Profile Usage Guide, see DSP1001

483 **4.20**

484 **REST**

Representational State Transfer, as originally and informally described in Architectural Styles and the Design of Network-based Software Architectures.

485 **4.21**

486 **SLP**

Service Location Protocol, defined by the IETF

487 **4.22**

488 **SNMP**

Simple Network Management Protocol, defined by the IETF

489 **4.23**

490 **UCS**

Universal Multiple-Octet Coded Character Set (short form: Universal Character Set)

491 **4.24**

492 **UML**

Unified Modeling Language, defined by the OMG

493

4.25

494   **URI**

Universal Resource Identifier, defined by the IETF

495   **4.26**

496   **W3C**

World Wide Web Consortium; see http://www.w3.org

497   **4.27**

498   **WBEM**

Web-Based Enterprise Management

499   **4.28**

500   **WBEM URI**

Web-Based Enterprise Management Universal Resource Identifier, as defined in DSP0207

501   **4.29**

502   **XML**

eXtensible Markup Language, defined by the W3C

503

ANNEX A
(informative)

Change log

504

Version	Date	Description
1.0.0	2014-08-28	

505

Bibliography

506   The following documents may provide additional background.

507   DMTF DSP0004, *CIM Infrastructure 2.8*,
      http://www.dmtf.org/standards/published_documents/DSP0004_2.8.pdf

508   DMTF DSP0200, *CIM Operations over HTTP 1.4,*
      http://www.dmtf.org/standards/published_documents/DSP0200_1.4.pdf

509   DMTF DSP0201, *Representation of CIM in XML 2.4*,
      http://www.dmtf.org/standards/published_documents/DSP0201_2.4.pdf

510   DMTF DSP0202, *CIM Query Language Specification 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0202_1.0.pdf

511   DMTF DSP0203, *DTD for Representation of CIM in XML 2.4*,
      http://www.dmtf.org/standards/published_documents/DSP0203_2.4.dtd

512   DMTF DSP0207, *WBEM URI Mapping 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0207_1.0.pdf

513   DMTF DSP0210, *CIM-RS Protocol 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0210_1.0.pdf

514   DMTF DSP0211, *CIM-RS Payload Representation in JSON 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0211_1.0.pdf

515   DMTF DSP0212, *Filter Query Language 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0212_1.0.pdf

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

517   DMTF DSP0226, *Web Services for Management Specification 1.1*,
      http://www.dmtf.org/standards/published_documents/DSP0226_1.1.pdf

518   DMTF DSP0227, *WS-Management CIM Binding Specification 1.1*,
      http://www.dmtf.org/standards/published_documents/DSP0227_1.1.pdf

519   DMTF DSP0228, *Message Registry XML Schema 1.2*,
      http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.2.xsd

520   DMTF DSP0230, *WS-CIM Mapping Specification 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf

521   DMTF DSP0231, *CIM Simplified Policy Language 1.0*,
      http://www.dmtf.org/standards/published_documents/DSP0231_1.0.pdf

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

523   DMTF DSP8020, *Metric Registry XML Schema Specification 1.1*,
      http://schemas.dmtf.org/wbem/metricregistry/1/dsp8020_1.1.xsd

524   DMTF DSP8028, *Management Profile XML Schema 1.1*,
      http://schemas.dmtf.org/wbem/mgmtprofile/1/dsp8028_1.1.xsd

525   DMTF DSP8044, *XSD for Representation of CIM in XML 2.4*,
      http://schemas.dmtf.org/wbem/cim-xml/2/dsp8044_2.4.xsd