



Web Services Resource Metadata 1.0 (WS-ResourceMetadataDescriptor)

Public Review Draft 01, June 27, 2006

Document identifier:

wsrf-ws_resource_metadata_descriptor-1.0-spec-pr-01

Location:

http://docs.oasis-open.org/wsrp/wsrp-ws_resource_metadata_descriptor-1.0-spec-pr-01.pdf

Artifact Type:

Specification

Technical Committee:

OASIS Web Services Resource Framework TC

Chair(s):

Dr. Ian Robinson, IBM
Dr. David Snelling, Fujitsu Laboratories of Europe

Editors:

Dan Jemiolo, IBM – danjemiolo@us.ibm.com

Abstract:

The components introduced by the WS Resource Framework (WSRF) address functional aspects of modeling stateful resources (such as systems resources) using Web services. WSRF uses WSDL (currently WSDL 1.1) as the form of service description. There is a need to be able to supplement the descriptive information available about a WS-Resource. The format of the information about the components of a WS-Resource is standardized by WSRF, most notably in the resource properties document [WS-ResourceProperties].

29 In the realm of resource properties, the loosely coupled operations for reading and writing
30 of properties [WS-ResourceProperties] would benefit from metadata. An example of this
31 type of metadata is the mutability constraints and an enumeration of possible values for
32 resource property elements. This document explains the need for such metadata and
33 proposes an information model representing it that would be applicable to Manageable
34 Resources and WS-Resources in general.

35

36 **Status:**

37 This document is published by this TC as a "public review draft".

38 This document was approved as a public review draft by the WSRF-TC on the above
39 date. The level of approval is also listed above. Check the current location noted above
40 for possible later revisions of this document.

41 Technical Committee members should send comments on this specification to the
42 Technical Committee's email list. Others should send comments to the Technical
43 Committee by using the "Send A Comment" button on the Technical Committee's web
44 page at www.oasis-open.org/committees/wsrf/.

45 For information on whether any patents have been disclosed that may be essential to
46 implementing this specification, and any offers of patent licensing terms, please refer to
47 the Intellectual Property Rights section of the Technical Committee web page
48 (www.oasis-open.org/committees/wsrf/ipr.php).

49 The non-normative errata page, if any, for this specification is located at [www.oasis-](http://www.oasis-open.org/committees/wsrf/)
50 [open.org/committees/wsrf/](http://www.oasis-open.org/committees/wsrf/).

51

Table of Contents

53	1 INTRODUCTION	5
54	1.1 GOALS AND REQUIREMENTS	6
55	1.1.1 Requirements	6
56	1.1.2 Non-Goals.....	6
57	1.2 TERMINOLOGY	7
58	1.3 NAMESPACES	8
59	2 TERMINOLOGY AND CONCEPTS	9
60	3 EXAMPLE	10
61	3.1 THE OPERATINGSYSTEM PORTTYPE.....	10
62	3.2 OPERATING SYSTEM PROPERTIES.....	11
63	3.2.1 Operating System Resource Property definitions	11
64	3.2.2 Identification Property definitions.....	11
65	3.2.3 MetadataDescriptor for Identification portType	12
66	3.2.4 MetadataDescriptor for OperatingSystem portType	13
67	4 LOGICAL MODEL FOR METADATA	16
68	5 INFORMATION MODEL FOR WS-RESOURCE METADATA	17
69	6 DEFINITIONS COMPONENT	18
70	6.1 METADATADESCRIPTOR COMPONENTS WITHIN A DEFINITIONS COMPONENT	19
71	7 METADATADESCRIPTOR COMPONENT	20
72	7.1 PROPERTIES COMPONENT OF A METADATADESCRIPTOR	21
73	8 PROPERTY COMPONENT	22
74	8.1 XML SCHEMA VALUE SPACE AND {VALIDVALUES}	24
75	8.2 VALIDVALUES	25
76	8.3 VALIDVALUERANGE.....	26
77	8.4 STATICVALUES	28
78	8.5 INITIALVALUES	29
79	9 DOCUMENTATION COMPONENT	31
80	10 OBTAINING A METADATADESCRIPTOR DOCUMENT	31
81	10.1 EXTENDING WSDL 1.1 PORTTYPE.....	31
82	10.2 USING RESOURE PROPERTY ELEMENTS TO EXPOSE METADATADESCRIPTORS.....	32
83	11 REFERENCES	34

84	11.1	NORMATIVE	34
85	11.2	NON-NORMATIVE.....	34
86	APPENDIX A. ACKNOWLEDGMENTS		35
87	APPENDIX B. XML SCHEMA FOR WS-RESOURCEMETADATADESCRIPTOR.....		36
88	APPENDIX C. REVISION HISTORY.....		45
89	APPENDIX D. NOTICES.....		46
90			

91 1 Introduction

92 In the WS-Resource Framework [WSRF], elements of a WS-Resource's state are exposed to
93 third party requestors through an XML document. The XML document associated with a WS-
94 Resource is called a *resource properties document*. The resource properties document is a
95 projection of the WS-Resource's state (not all of the elements of a WS-Resource's state are
96 exposed through the resource properties document). An individual element of state contained in a
97 resource properties document is called a *resource property*. Access to the resource properties
98 document is governed by Web services operations defined in the WS-Resource Properties [WS-
99 Resource Properties] specification. These operations generically allow for get, set and query of
100 resource properties.

101

102 In many cases, some of the resource properties exposed through the resource properties
103 document are not accessible through every operation defined in WS-ResourceProperties. The
104 most common case of this is a resource property that is "read-only" implying that a requestor may
105 not use Web services message exchanges (such as the WS-ResourceProperties
106 SetResourceProperties operation) to change the value of the resource property. Clearly, an
107 implementation of a WS-Resource is likely to return a fault message if a requestor attempts to
108 change the value of a "read-only" resource property. However, in the absence of additional
109 metadata, there is no standard means by which the requestor can determine a priori that the
110 resource property was not modifiable.

111

112 We refer to the concept of a WS-Resource Metadata Descriptor to describe a unit of metadata
113 information associated with the interface components of a WS-Resource. We describe an
114 information model that outlines the components of metadata and their relationships to interface
115 description artifacts such as WSDL 1.1 portTypes and resource properties document schema
116 definitions.

117

118 A WS-Resource Metadata Descriptor serves multiple purposes. The first is to provide additional
119 information about the resource properties of a WS-Resource. For instance, indicating whether a
120 resource property is changeable using Web services message exchanges such as the
121 SetResourceProperties operation described in the WS-ResourceProperties specification [WS-
122 ResourceProperties]. This aspect of the MetadataDescriptor is associated with the interface of
123 the WS-Resource, and would not vary between different implementations of the interface.
124 Information in the MetadataDescriptor provides clients of a WS-Resource the potential for greater
125 understanding of the behavior of that WS-Resource.

126

127 The second purpose is to provide information about the value restrictions of the resource
128 properties in the resource properties document for the WS-Resource. This additional information
129 may be associated with implementations of the interface as well as with the WSDL interface
130 definition.

131 The single portType that describes the manageability interface for a manageable resource type is
132 derived from various other manageability portTypes. With WSDL 1.1, physically including, using
133 copy and paste, the operations from each of these portTypes into the definition of the most
134 derived portType achieves this inheritance. Each of the portTypes from which a manageable
135 resource's portType is derived may also have a MetadataDescriptor to augment its description.
136 Each of the portTypes may have an optional attribute information item that references a
137 MetadataDescriptor component by its namespace qualified name (QName).

138

139 This document standardizes the form of the WS-Resource MetadataDescriptor that contains
140 metadata information about a WS-Resource's interface so that clients of that interface may
141 reason about implementations of the interface at both design time and run time. The syntax of a
142 preferred XML serialization of the information model is also described.

143

144 A portion of the Global Grid Forum's "Open Grid Services Infrastructure (OGSI) Version 1.0"
145 specification [OGSI] inspired many of the concepts expressed in this document.

146

147 **1.1 Goals and Requirements**

148 The goal of this document is to define the terminology, concepts, information model and XML
149 definitions needed to express the metadata requirements of WS-Resources, as defined by the
150 [WS-Resource] specification.

151

152 **1.1.1 Requirements**

153 In meeting this goal, the specification MUST address the following specific requirements:

- 154 • Define an information model representing metadata about resource properties associated
155 with a WS-Resource interface.
- 156 • Define a standard annotation for associating metadata descriptions with other description
157 artifacts of the WS-Resource, particularly its WSDL 1.1 portType and its resource
158 properties document definition.
- 159 • Define the standard schema for representing the aspects of the information model.

160

161 **1.1.2 Non-Goals**

162 The following topics are outside the scope of this specification:

- 163 • It is not an objective of this specification to define new message exchanges required to
164 access the metadata from a WS-Resource.
- 165 • It is not an objective of this specification to describe the means required to store the
166 metadata for a WS-Resource.

167 1.2 Terminology

168 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
169 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
170 interpreted as described in [RFC 2119].

171

172 When describing abstract data models, this specification uses the notational convention used by
173 the [XML Infoset]. Specifically, abstract property names always appear in square brackets (e.g.,
174 [some property]).

175

176 This specification uses a notational convention, referred to as "Pseudo-schemas" in a fashion
177 similar to the WSDL 2.0 Part 1 specification [WSDL 2.0]. A Pseudo-schema uses a BNF-style
178 convention to describe attributes and elements:

179 `?' denotes optionality (i.e. zero or one occurrences),

180 `*' denotes zero or more occurrences,

181 `+' one or more occurrences,

182 `[` and `]' are used to form groups,

183 `|' represents choice.

184

185 Attributes are conventionally assigned a value which corresponds to their type, as defined in the
186 normative schema.

187

```
188 <!-- sample pseudo-schema -->  
189 <element  
190     required_attribute_of_type_QName="xs:QName"  
191     optional_attribute_of_type_string="xs:string"? >  
192     <required_element />  
193     <optional_element />?  
194     <one_or_more_of_these_elements />+  
195     [ <choice_1 /> | <choice_2 /> ]*  
196 </element>
```

197

198

199

200

201

202

203

1.3 Namespaces

204

The following namespaces are used in this document:

Prefix	Namespace
xs	http://www.w3.org/2001/XMLSchema
wsdl	http://schemas.xmlsoap.org/wsdl
wsrf-rp	http://docs.oasis-open.org/wsrp/rp-2
wsrmd	http://docs.oasis-open.org/wsrp/rmd-1

205

206 **2 Terminology and Concepts**

207 The following definitions outline the terminology and usage in this specification. This section gives
208 only brief description of these terms.

209

210 Metadata:

- 211 • Data about data. In practice, metadata comprises a structured set of descriptive elements
212 to describe an information resource. Currently, only a WS-Resource's resource
213 properties have metadata definitions.

214

215 MetadataDescriptor:

- 216 • A unit of containment for resource property metadata for a WS-Resource's interface;
217 property metadata is defined by zero or more Property elements.

218

219 MetadataDescriptor Document:

- 220 • An XML instance document whose root is a Definitions element from the wsrmd
221 namespace. This document contains definitions for zero or more MetadataDescriptor
222 components.

223 3 Example

224 In the following example there are “Operating System” WS-Resources whose values are
225 projected from the implementation of the *OperatingSystem* portType.

226 3.1 The OperatingSystem portType

227 The *OperatingSystem* portType defines operations and a resource properties document that
228 describes the Web services interface to operating system resource instances. As well as
229 providing a mechanism for interacting with the operating system itself, this portType also
230 describes properties, which represent the hardware of the underlying machine on which the
231 operating system is running.

232

233 The *OperatingSystem* portType is derived from various other manageability portTypes – these
234 illustrations use some of the function from the *Identification* manageability portType. As is
235 required with WSDL 1.1, this derivation is achieved by physically including the definitions from
236 each of these portTypes in the definition of the *OperatingSystem* portType. This “cut-and-paste”
237 system of derivation is discussed further in [AppNotes].

238

239 The *OperatingSystem* portType is sketched as follows:

240

```
241 (01) ... xmlns:os="http://example.com/ns/OperatingSystem"  
242 (02) ... xmlns:id="http://example.com/ns/Identification"  
243 (03) ...  
244 (04) <portType name="OperatingSystem"  
245 (05)     wsrf-rp:ResourceProperties="os:OSResourceProperties"  
246 (06)     ..wsrmd:Descriptor="os:OperatingSystemMetadataDescriptor"  
247 (07)     ..wsrmd:DescriptorLocation="http://example.com/OperatingSystem.wsrmd" >  
248 (08) ...  
249 (09) </portType>
```

250

251 Line (04) contains a portType declaration for a portType named *OperatingSystem* in the
252 namespace corresponding to the *os:* namespace prefix declaration.

253

254 Line (05) indicates the global XML element declaration of the root element of the resource
255 properties document associated with any WS-Resource whose Web service implements the
256 *os:OperatingSystem* portType.

257

258 Line (06) identifies that a *MetadataDescriptor* has been defined for this interface identified by the
259 QName *os:OperatingSystemMetadataDescriptor*.

260

261 Line (07) - (07) indicate that information about MetadataDescriptors in the namespace
262 corresponding to the os: namespace prefix declaration can be found by dereferencing
263 <http://example.com/OperatingSystem.wsrmd>.

264 3.2 Operating System Properties

265 In the following example we define a subset of the resource properties of the operating system.
266 Following that are the MetadataDescriptors for the Identification and Operating System
267 portTypes.

268 3.2.1 Operating System Resource Property definitions

```
269 (10) ...  
270 (11)   xmlns:os="...  
271 (12)  
272 (13) <element name="OSResourceProperties">  
273 (14)   <complexType>  
274 (15)     <sequence>  
275 (16)       <element ref="id:ResourceType" minOccurs="0" maxOccurs="1"/>  
276 (17)       <element ref="id:ResourceID" minOccurs="0" maxOccurs="1"/>  
277 (18)       <element ref="os:numberOfProcesses" minOccurs="0" maxOccurs="1"/>  
278 (19)       <element ref="os:totalSwapSpaceSize" minOccurs="0" maxOccurs="1"/>  
279 (20)       <element ref="os:processor" minOccurs="1" maxOccurs="unbounded"/>  
280 (21)     </sequence>  
281 (22)   </complexType>  
282 (23) </element>  
283 (24)  
284 (25) <element name="numberOfProcesses" type="xsd:int" />  
285 (26) <element name="totalSwapSpaceSize" type="xsd:unsignedLong" />  
286 (27) <element name="processor" type="xsd:string" />  
287 (28)
```

288 3.2.2 Identification Property definitions

289

```
290 (29) <element name="IdentificationResourceProperties">  
291 (30)   <complexType>  
292 (31)     <sequence>  
293 (32)       <element ref="ResourceType" minOccurs="0" maxOccurs="1"/>  
294 (33)       <element ref="ResourceID" minOccurs="0" maxOccurs="1"/>  
295 (34)     </sequence>  
296 (35)   </complexType>  
297 (36) </element>
```

298
299
300

```
(37)
(38) <element name="ResourceType" type="xs:string" />
(39) <element name="ResourceID" type="xs:string" />
```

301 3.2.3 MetadataDescriptor for Identification portType

302 An example MetadataDescriptor document for the Identification portType is included below. For
303 the purposes of this example, this MetadataDescriptor document will be located at the URL

304 <http://example.com/metadataDescriptors/Identification.wsrmtd>.

305

306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

```
(40)
(41) <Definitions
(42)     xmlns="http://docs.oasis-open.org/wsr/1.0/"
(43)     xmlns:id="http://example.com/ns/Identification"
(44)     targetNamespace="http://example.com/ns/Identification">
(45)   <MetadataDescriptor
(46)     name="IdentificationMetadataDescriptor"
(47)     interface="id:Identification"
(48)     wsdlLocation="http://example.com/ns/Identification
(49)       http://example.com/wsd/Identification.wsdl" >
(50)   <Property name="id:ResourceID"
(51)     mutability="constant"
(52)     modifiability="read-only" />
(53)   <Property name="id:ResourceType"
(54)     mutability="constant"
(55)     modifiability="read-only" />
(56)   </MetadataDescriptor>
(57) </Definitions>
```

324

325 Line (41) contains a Definitions element defining MetadataDescriptor elements for the target
326 namespace identified in line (44).

327

328 There is one MetadataDescriptor element child of this Definitions element (lines (45)–(56)).

329 The name of the MetadataDescriptor is contained in line (46). This together with the namespace
330 prefix declaration in line (43) corresponding to the targetNamespace of the Definitions element
331 means the QName of the MetadataDescriptor is id:IdentificationMetadataDescriptor.

332

333 Line (47) identifies the QName of the portType (interface) with which this MetadataDescriptor is
334 associated. The location of WSDL for the Identification portType is expressed in the wsdlLocation
335 attribute in lines (48)-(49). This follows the pattern of the wsdl:wsdlLocation attribute defined in
336 the WSDL 2.0 specification [WSDL 2.0].

337

338 Lines (50)-(55) show two Property elements containing metadata information about resource
339 properties defined in the resource properties document for the Identification portType. Lines (50)-
340 (52) contain the first Property element that references the QName of the id:ResourceID resource
341 property. Line (51) indicates that the id:ResourceID resource property element will always have a
342 constant value. Line (52) states that the id:ResourceID resource property is read-only, meaning
343 that it cannot be changed by a requestor using Web services message exchanges such as the
344 SetResourceProperties operation as defined in WS-ResourceProperties.

345

346 The second Property element, in lines (53)-(55), references the QName of the id:ResourceType
347 resource property in line (53). This resource property element has the same metadata attributes
348 as id:ResourceIdentifier.

349 **3.2.4 MetadataDescriptor for OperatingSystem portType**

350 For the purposes of this example the MetadataDescriptor document for the OperatingSystem
351 portType is found at <http://example.com/metadataDescriptors/OperatingSystem.wsrm>.
352

353 The contents of the MetadataDescriptor for the OperatingSystem portType appear as follows:
354

```
355 (58)      <Definitions
356 (59)          xmlns="http://docs.oasis-open.org/wsrf/rmd-1"
357 (60)          xmlns:id="http://example.com/ns/Identification"
358 (61)          xmlns:os="http://example.com/ns/OperatingSystem"
359 (62)          targetNamespace="http://example.com/ns/OperatingSystem">
360 (63)      <MetadataDescriptor
361 (64)          name="OperatingSystemMetadataDescriptor"
362 (65)          interface="os:OperatingSystem"
363 (66)          wsdlLocation="http://example.com/ns/OperatingSystem
364 (67)              http://example.com/wsdl/OperatingSystem.wsdl">
365 (68)      <Property name="id:ResourceID"
366 (69)          mutability="constant"
367 (70)          modifiability="read-only" />
368 (71)      <Property name="id:ResourceType"
369 (72)          mutability="constant"
370 (73)          modifiability="read-only">
371 (74)          <ValidValues>
372 (75)              <id:ResourceType>SuSELinux</id:ResourceType>
373 (76)              <id:ResourceType>IBMzOS</id:ResourceType>
374 (77)              <id:ResourceType>MicrosoftWindows_XP</id:ResourceType>
375 (78)          </ValidValues>
376 (79)      </Property>
```

```

377 (80)         <Property name="os:numberOfProcesses"
378 (81)             mutability="mutable"
379 (82)             modifiability="read-only" />
380 (83)         <Property name="os:processor"
381 (84)             mutability="constant"
382 (85)             modifiability="read-only">
383 (86)             <ValidValues>
384 (87)                 <os:processor>Pentium Family</os:processor>
385 (88)                 <os:processor>Power PC 750</os:processor>
386 (89)                 <os:processor>68xxxx Family</os:processor>
387 (90)                 <os:processor>PA-RISC Family</os:processor>
388 (91)                 <os:processor>Alpha Family</os:processor>
389 (92)                 <os:processor>IBM390 Family</os:processor>"
390 (93)                 <os:processor>G5</os:processor>
391 (94)                 <os:processor>AMD</os:processor>
392 (95)             </ValidValues>
393 (96)         </Property>
394 (97)     </MetadataDescriptor>
395 (98) </Definitions>

```

396 Lines (58) to (98) contain a Definitions element for the <http://example.com/ns/OperatingSystem>
397 namespace.

398

399 Lines (63) to (98) contain the definition of the MetadataDescriptor with QName
400 `os:OperatingSystem MetadataDescriptor`.

401

402 Line (65) indicates that this MetadataDescriptor corresponds to the OperatingSystem portType.

403 Lines (66)-(67) gives the location of the WSDL document that defines elements associated with
404 the namespace URI associated with the `os:` prefix (i.e. the WSDL definitions element that defines
405 the OperatingSystem portType).

406

407 Lines (68)-(96) contains the four properties described in this MetadataDescriptor example.

408 Lines (68) –(70) contain the ResourceID Property element copied from the
409 `id:IdentificationMetadataDescriptor` describing the Identification portType.

410 Lines (71)-(79) contain the Property element describing the `id:ResourceType` resource property
411 from the Identification portType. Lines (74)-(78) contain the set of ValidValues that the
412 `id:ResourceType` resource property may contain.

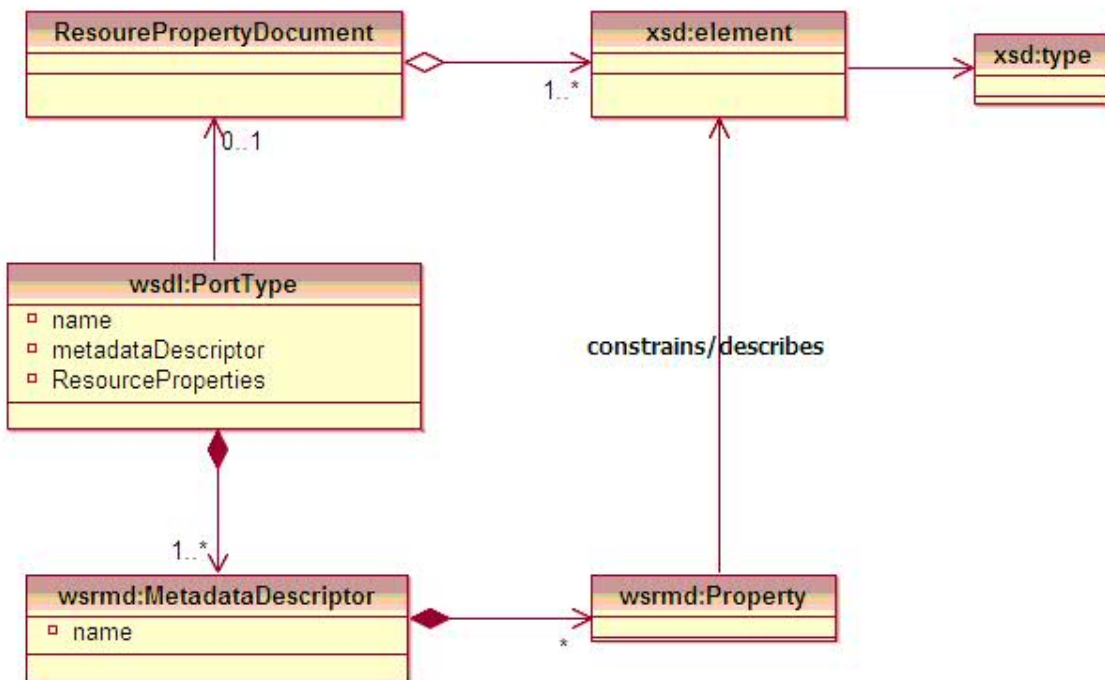
413

414 Lines (80)-(82) contain the `os:numberOfProcesses` Property element which references the
415 QName of the `os:numberOfProcesses` resource property. Line (81) indicates that the value of the
416 `os:numberOfProcesses` may change over time. Line (82) indicates that, the

417 os:numberOfProcesses can not be changed by a requestor using Web services message
418 exchanges such as the SetResourceProperties operation as defined in WS-ResourceProperties
419 [WS-ResourceProperties].
420 The next Property element references the os:processor. The modifiability and mutability values
421 indicate that the property is static – it will not change during the resource’s lifetime. Lines (87)-
422 (94) describe valid values for the os:processor.

423 **4 Logical Model for Metadata**

424 The following figure shows a logical model depicting the relationship between the various
 425 elements of metadata description and those elements the metadata describes.



426

427 *Figure 1 Logical Model of WS-Resource MetadataDescriptor*

428

429 In our model, the unit of metadata containment is referred to as a *MetadataDescriptor*. A
 430 *MetadataDescriptor* is used to describe aspects of a WS-Resource's interface, particularly those
 431 elements associated with the WS-Resource's WSDL 1.1 portType.

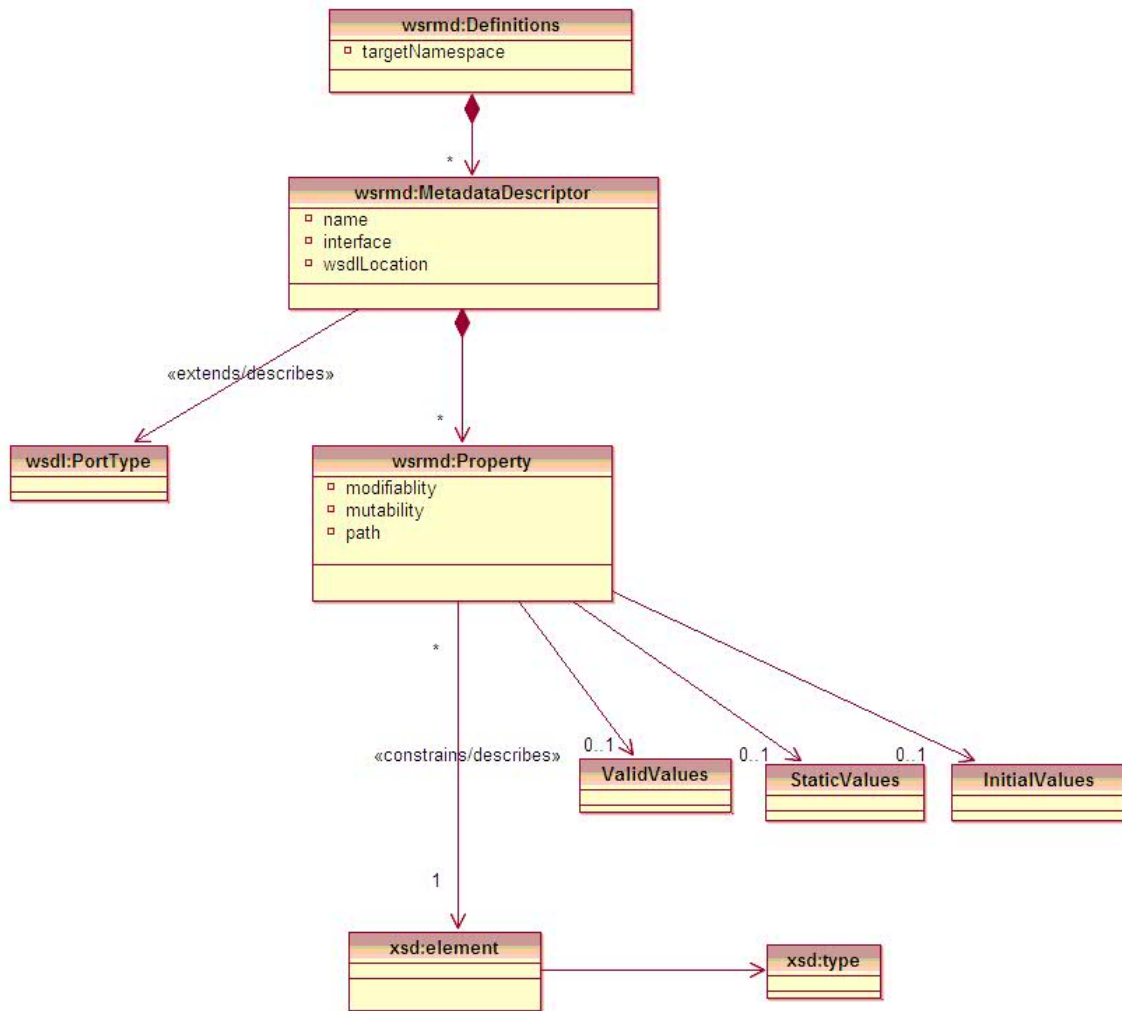
432

433 A *MetadataDescriptor* contains metadata describing and/or constraining resource property
 434 elements as contained within a WS-Resource's Resource Properties document type definition.
 435 Each resource property element is defined as an XML Schema global element, in some
 436 namespace.

437
438
439
440
441
442

5 Information Model for WS-Resource Metadata

This section describes the information model for metadata describing/constraining the resource properties of WS-Resources. The model is a simple hierarchy – each WSDL portType *references* a Resource Metadata Descriptor document, and that file *contains* a Definitions element, which *contains* MetadataDescriptors, which *contain* Property elements. A UML diagram of this model is shown in the following figure:



443
444
445

Figure 2 - Information Model for WS-Resource Metadata Descriptor

446
447

We describe the Definitions, MetadataDescriptor, and Property components in the following sections.

448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484

6 Definitions Component

The Definitions component is a container for a set of MetadataDescriptor components (see section 7). The Definitions component defines a targetNamespace which forms the {namespace} property of all components it contains.

The properties of a Definitions component are as follows:

- {targetNamespace} a namespace URI that applies as the {namespace} property to all [child] components.
- {metadataDescriptors} a set of zero or more MetadataDescriptor components.

The following is an XML representation of the Definitions component:

```
<Definitions
  targetNamespace="xs:anyURI"
  {anyAttribute}* >

  <documentation />?
  <MetadataDescriptor /> *

</Definitions>
```

The Definitions *element information item* has the following Infoset properties:

- A [local name] of "Definitions".
- A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- one or more *attribute information items* amongst its [attributes] as follows:
 - A REQUIRED targetNamespace *attribute information item*
 - The value of this attribute information item contains a URI that defines the {namespace} property of all [child] components.
 - The type of the targetNamespace *attribute information item* is xs:anyURI.
 - Zero or more namespace qualified *attribute information items*. The [namespace name] of such *attribute information items* MUST NOT be "http://docs.oasis-open.org/wsrf/rmd-1".
- Zero or more *element information items* amongst its [children], in order as follows:
 - An OPTIONAL documentation *element information item* (See section 7).
 - Zero or more MetadataDescriptor *element information items* (See section 6.1).

485 **6.1 MetadataDescriptor components within a Definitions**
486 **component**

487 All MetadataDescriptor components defined in a given namespace MUST appear as [children] of
488 a Definitions component with {targetNamespace} value the same URI as that namespace. All
489 MetadataDescriptor components MUST be uniquely named, implying that the {name} property of
490 the MetadataDescriptor component MUST be unique amongst the {metadataDescriptors} of a
491 Definitions component.

492 7 MetadataDescriptor Component

493 The MetadataDescriptor component is a container for a set of metadata descriptions and
494 constraints on a WS-Resource. The MetadataDescriptor component contains additional
495 information that describes or constrains various aspects of a WS-Resource. For example, it
496 provides additional information about the interface of the WS-Resource relevant to the
497 management of the resource. In particular, it allows tools and applications, such as management
498 applications, to be able to reason in detail about the WS-Resource both at runtime and at
499 development time when no instances of the WS-Resource are available.

500

501 The properties of a MetadataDescriptor component are as follows:

- 502 • {name} a name of a MetadataDescriptor component.
- 503 • {namespace} a namespace URI of the MetadataDescriptor component.
- 504 • {QName} a combination of the {name} and {namespace} of the MetadataDescriptor
505 component.
- 506 • {interface} a QName identifying a Web services interface definition with which this
507 MetadataDescriptor is associated.
- 508 • {wsdlLocation} a set of URI pairs, each pair associating a namespace URI with a URL of
509 a document containing a WSDL definition of that namespace. This is a similar
510 mechanism to that used in WSDL 2.0 [WSDL2.0].
- 511 • {properties} A set of zero or more Property components

512

513 The following is an XML representation of the MetadataDescriptor component:

514

```
515 <MetadataDescriptor  
516     name="xs:NCName"  
517     interface="xs:QName"  
518     wsdlLocation="list of xs:anyUri?"  
519     {anyAttribute}* >  
520 <documentation /> ?  
521 <Property /> *  
522 {any}*  
523 </MetadataDescriptor>
```

524

525

526 The MetadataDescriptor *element information item* has the following Infoset properties:

- 527 • A [local name] of "MetadataDescriptor" .
- 528 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".

- 529
- two or more *attribute information items* amongst its [attributes] as follows:
 - 530 ○ A REQUIRED name attribute information item
 - 531 ▪ The value of this *attribute information item* contains the name of this
 - 532 MetadataDescriptor component.
 - 533 ▪ The type of the name *attribute information item* is xs:NCName.
 - 534 ○ A REQUIRED interface attribute information item
 - 535 ▪ The value of this *attribute information item* contains a QName of a WSDL
 - 536 1.1 portType element or WSDL 2.0 interface element associated with this
 - 537 MetadataDescriptor component.
 - 538 ▪ The type of the interface *attribute information item* is xs:QName.
 - 539 ○ An OPTIONAL wsdlLocation attribute information item
 - 540 ▪ The value of this *attribute information item* contains a list of pairs of
 - 541 URIs; where the first URI of the pair, which MUST be an absolute URI as
 - 542 defined in [URI], indicates a WSDL namespace name, and, the second a
 - 543 hint as to the location of a WSDL document defining WSDL components
 - 544 for that namespace name. The second URI of a pair MAY be absolute or
 - 545 relative.
 - 546 ▪ The type of the wsdlLocation *attribute information item* is list of
 - 547 xs:anyURI.
 - 548 ○ Zero or more namespace qualified *attribute information items*. The [namespace
 - 549 name] of such *attribute information items* MUST NOT be "http://docs.oasis-
 - 550 open.org/wsrf/rmd-1".
 - 551 • Zero or more *element information items* amongst its [children], in order as follows:
 - 552 ○ An OPTIONAL documentation *element information item* (See section 9).
 - 553 ○ Zero or more Property *element information items* (See section 8)
 - 554 ○ Zero or more namespace-qualified *element information items*. The [namespace
 - 555 name] of such *element information items* MUST NOT be "http://docs.oasis-
 - 556 open.org/wsrf/rmd-1".

557 7.1 Properties component of a MetadataDescriptor

558 The {properties} of a MetadataDescriptor contains a set of Property components, defining
559 additional metadata and constraints on resource property elements (and attributes) associated
560 with a MetadataDescriptor. The definition of a Property component's scope is contained in
561 Section **Error! Reference source not found.**

562 8 Property Component

563 The Property component is a container for a set of metadata descriptions and constraints on a
564 specific Resource Property element or attribute thereof. The properties of a Property component
565 are as follows:

- 566 • {name} an identifier of the XML element to which the Property component applies. This is
567 defined as a resource property's QName.
- 568 • {mutability} an xs:string enumeration of "constant", "appendable", or "mutable".
- 569 • {modifiability} an xs:string enumeration of "read-only" or "read-write".
- 570 • {subscribability} an xs:boolean indicating, if true, that the Resource Property element
571 associated with the {name} can be the target of a subscription.
- 572 • {validValues} optional choice of one ValidValues component or one ValidValueRange
573 component.
- 574 • {staticValues} optional choice of one StaticValues component.
- 575 • {initialValues} optional choice of one InitialValues component.
- 576 • {attributes} zero or more Attribute components.

577

578 A Property component MAY also contain additional "extension" components added using the
579 extensibility mechanism defined by this specification. The following is an XML representation of
580 the Property component:

581

```
582 <Property  
583     name="xs:QName"  
584     mutability="[constant|appendable|mutable]" ?  
585     modifiability="[read-only|read-write]" ?  
586     subscribability="xs:boolean" ?  
587     {anyAttribute}* >  
588  
589     <documentation />?  
590     [ <ValidValues> {any}* </ValidValues> |  
591       <ValidValueRange  
592         lowerBound="xs:anySimpleType"? upperBound="xs:anySimpleType"?  
593     /> ] ?  
594     <StaticValues> {any}* </StaticValues> ?  
595     <InitialValues> {any}* </InitialValues> ?  
596     {any}*  
597 </Property>
```

598

599 The Property *element information item* has the following Infoset properties:

- 600
- A [local name] of "Property" .
- 601
- A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 602
- one or more *attribute information items* amongst its [attributes] as follows:
- 603
- A REQUIRED name attribute information item
- 604
- The value of this *attribute information item* MUST contain a QName of
- 605
- the Resource Property (an XML Schema global element definition)
- 606
- contained within the Resource Properties document associated with the
- 607
- portType or interface identified by {interface}. The Resource Property
- 608
- element MUST conform to the requirements specified for Resource
- 609
- Property declarations in WS-ResourceProperties.
- 610
- An OPTIONAL mutability attribute information item
- 611
- The value of this *attribute information item* expresses how the value of
- 612
- the {name} can change over time.
- 613
- The type of the mutability *attribute information item* is an xs:string
- 614
- restricted to the following enumeration:
- 615
- "constant"
- 616
- The values of the {name} MUST NOT change after WS-
- 617
- Resource creation.
- 618
- "mutable"
- 619
- The values of the {name} MAY change at any time during
- 620
- the lifetime of the WS-Resource. Existing values MAY be
- 621
- removed and new values MAY be added.
- 622
- "appendable"
- 623
- The values of the {name} MAY have new values added
- 624
- during the lifetime of the WS-Resource. Once added those
- 625
- values MUST NOT be removed.
- 626
- If the mutability *attribute information item* is not defined, the value of the
- 627
- mutability property is "unknown".
- 628
- An OPTIONAL modifiability *attribute information item*
- 629
- The value of this *attribute information item* indicates whether a requestor
- 630
- can modify the value of the {name}.
- 631
- The type of the modifiability *attribute information item* is an xs:string
- 632
- restricted to the following enumeration:
- 633
- "read-only" – The value of the {name} can not be changed by
- 634
- Web services message exchanges such as the
- 635
- SetResourceProperty message as defined in WS-
- 636
- ResourceProperties.
- 637
- "read-write" – The value of the {name} MAY be changed by Web
- 638
- services message exchanges such as the SetResourceProperty

639 message as defined in WS-ResourceProperties. Note - If the
640 value of the modifiability *attribute information item* is “read-write”
641 then the value of the mutability *attribute information item* MUST
642 NOT be “constant”.

- 643 ▪ If the modifiability attribute information item is not defined, the value of
644 the modifiability property is “unknown”.
- 645 ○ An OPTIONAL subscribability *attribute information item*
646 ▪ The value of this *attribute information item* expresses whether the
647 Resource Property element associated with the {name} can be the target
648 of a subscription. The default value is “false”. Note: The actual
649 subscription semantics are dependent on whatever notification
650 mechanism, if any, (such as WS-BaseNotification [WS-BaseNotification])
651 is supported.
- 652 ○ Zero or more namespace qualified *attribute information items*. The [namespace
653 name] of such *attribute information items* MUST NOT be "http://docs.oasis-
654 open.org/wsrf/rmd-1".
- 655 • Zero or more *element information items* amongst its [children], in order as follows:
656 ○ An OPTIONAL documentation *element information item* (See section 9).
657 ○ An OPTIONAL *element information item* from among the following:
658 ○ A ValidValues element information item (See section 8.2)
659 ○ A ValidValueRange element information item (See section 8.3)
660 ○ An OPTIONAL StaticValues element information item (See section 8.4)
661 ○ An OPTIONAL InitialValues element information item (See section 8.5)
662 ○ Zero or more namespace-qualified *element information items*. The [namespace
663 name] of such *element information items* MUST NOT be "http://docs.oasis-
664 open.org/wsrf/rmd-1".

665 **8.1 XML Schema value space and {validValues}**

666 When creating a resource property (ie defining an XML Global Element), the XML Schema
667 designer defines the semantic of the property and uses XML Schema to express the value space
668 of the resource property (based on the semantics of the property) and all of its descendant
669 *element information items* and *attribute information items*. This is a different concept from what is
670 expressed by defining the {validValues}. When specifying {validValues} in a metadata description,
671 one does not redefine the semantic of the {name} nor its value space. Specifying {validValues}
672 expresses constraints on the value space that are appropriate for the specific use of the {name}.
673 This distinction should guide designers in deciding whether to use XML Schema mechanisms or
674 a metadata description to restrict value space of a {name}. The value space defined by
675 {validValues} for a {name} MUST be contained within the XML Schema definition of the {name}.

676 8.2 ValidValues

677 The purpose of the ValidValues component is to restrict the set of valid values that a [parent]
678 Property component's {name} may contain.

679

680 If the {validValues} of a Property component is not empty, and contains a ValidValues description,
681 then any Web service that implements the portType or interface identified by {interface} MUST
682 ensure that the value(s) of the {name} of the [parent] Property component MUST correspond to
683 one of the values enumerated within the set of {validValues}.

684

685 Note: because the child *element information items* of a ValidValues *element information item* are
686 XML fragments, it is not required that these fragments be validated (processContents is "skip"). .

687 The properties of a ValidValues component are as follows.

688

- 689 • {values} zero or more XML fragments that correspond to the type of the [parent] Property
690 component's {name}.

691

692 The following is an XML representation of the ValidValues component:

693

694

```
<ValidValues  
  {anyAttribute}* >  
  <documentation />?  
  {any}*  
</ValidValues>
```

695

696

697

698

699

700 The ValidValues *element information item* has the following Infoset properties:

701

- 701 • A [local name] of "ValidValues".
- 702 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 703 • zero or more *attribute information items* amongst its [attributes] as follows:
 - 704 ○ Zero or more namespace qualified *attribute information items*. The [namespace
705 name] of such *attribute information items* MUST NOT be "http://docs.oasis-
706 open.org/wsrf/rmd-1".
- 707 • Zero or more *element information items* amongst its [children], either:
 - 708 ○ An OPTIONAL documentation *element information item* (See section 9).
 - 709 ○ Zero or more namespace-qualified *element information items*. The [namespace
710 name] of such *element information items* MUST NOT be "http://docs.oasis-
711 open.org/wsrf/rmd-1".

- 712 ▪ Each *element information item* MUST be an XML fragment that
713 corresponds to the type of the XML element identified by the [parent]
714 Property component's {name}
715 ▪ Note, because these are XML fragments, it is not expected that a
716 processor of a MetadataDescriptor document would need to validate
717 these element information items (processContents = "skip").
718 ○ Zero or more character information items.
719

720 **8.3 ValidValueRange**

721 The ValidValueRange component is an alternative mechanism to specify the set of ValidValues
722 for the [parent] Property component's {name}. Unlike the ValidValues component, which specifies
723 an enumeration of values, the ValidValueRange restricts the set of valid values for a {name} by
724 specifying a range of possible values. This mechanism can only be used when the {name} is an
725 XML element of simpleType.

726 ValidValueRange defines an optional inclusive lower bound of the range and optional inclusive
727 upper bound of the range. Both MAY be specified. At least one MUST be specified. The values of
728 the lower bound and upper bound (if specified) MUST correspond to the value space definition of
729 the {name}. If the {lowerBound} of this attribute information is NOT specified, its default value is
730 defined by the lowest possible value defined for the value space of the {name} or "undefined".
731 Similarly the default value of {upperBound} is the largest value for the value space of the {name}
732 or "undefined".
733

734 If the {validValues} of a Property component is not empty and contains a ValidValueRange
735 description, then any Web service that implements the portType or interface identified by
736 {interface} MUST ensure that the value(s) of the resource property as identified by the {name} of
737 the Property component MUST correspond to a value within the range specified by {validValues}.
738

739 The properties of a ValidValueRange component are as follows:

- 740 • {lowerBound} the (inclusive) lower bound of the value space defined by this component
741 for the [parent] Property component's {name}.
- 742 • {upperBound} the (inclusive) upper bound of the value space defined by this component
743 for the [parent] Property component's {name}.
- 744 • {range} a range of values, bounded by the values of {lowerBound} and {upperBound}.
745 The values within {range} MUST be compliant with any value space constraints specified
746 on the type definition of the [parent] Property component's {name}.

747

748 The following is an XML representation of the ValidValues component:
749

```

750 <ValidValueRange
751     lowerBound="xs:anySimpleType" ? upperBound="xs:anySimpleType" ?
752     {anyAttribute}* >
753     <documentation />?
754     {any}*
755 </ValidValueRange>

```

756

757 The ValidValueRange *element information item* has the following Infoset properties:

- 758
- 759 • A [local name] of "ValidValueRange" .
 - 760 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
 - 761 • one or more *attribute information items* amongst its [attributes] as follows:
 - 762 ○ one or more attribute information items amongst:
 - 763 ▪ An OPTIONAL lowerBound attribute information item
 - 764 • The value of this *attribute information item* defines an inclusive lower bound on the range of valid values to apply to the [parent] Property component's {name}.
 - 765 • The type of the lowerBound *attribute information item* is an xs:anySimpleType. This type MUST correspond to the type of the [parent] Property component's {name}.
 - 766 • The value of this *attribute information item* MUST conform to any value space constraints specified on the type definition of the [parent] Property component's {name}.
 - 767 ▪ An OPTIONAL upperBound attribute information item
 - 768 • The value of this *attribute information item* defines an inclusive upper bound on the range of valid values to apply to the [parent] Property component's {name}.
 - 769 • The type of the upperBound *attribute information item* is an xs:anySimpleType. This type MUST correspond to the type of the [parent] Property component's {name}.
 - 770 • The value of this *attribute information item* MUST conform to any value space constraints specified on the type definition of the [parent] Property component's {name}.
 - 771 ○ Zero or more namespace qualified *attribute information items*. The [namespace name] of such *attribute information items* MUST NOT be "http://docs.oasis-open.org/wsrf/rmd-1".
 - 772 • Zero or more *element information items* amongst its [children], in order as follows:
 - 773 ○ An OPTIONAL documentation *element information item* (See section 9).
- 774
- 775
- 776
- 777
- 778
- 779
- 780
- 781
- 782
- 783
- 784
- 785
- 786

787 o Zero or more namespace-qualified *element information items*. The [namespace
788 name] of such *element information items* MUST NOT be "http://docs.oasis-
789 open.org/wsrf/rmd-1".
790

791 **8.4 StaticValues**

792 The purpose of the StaticValues component is to define the minimum set of values that a [parent]
793 Property component's {name} must contain.

794

795 If the {staticValues} of a Property component is not empty, any Web service that implements the
796 portType or interface identified by {interface} MUST ensure that all the value(s) defined in
797 {staticValues} appear in the {name}.

798

799 The values contained in a StaticValues component MUST conform to the XML Schema definition
800 of the {name}. Note: because the child *element information items* of a StaticValues *element*
801 *information item* are XML fragments, it is not required that these fragments be validated
802 (processContents is "skip").

803

804 The properties of a StaticValues component are as follows:

- 805 • {values} zero or more XML fragments that correspond to the type of the [parent] Property
806 component's {name}.
- 807 • The number of XML fragments within {values} MUST NOT be greater than the
808 maxOccurs facet of the schema declaration target of the {name}.

809

810 The following is an XML representation of the StaticValues component:

811

```
812 <StaticValues  
813     {anyAttribute}* >  
814     <documentation />?  
815     {any}*  
816 </StaticValues>
```

817

818 The StaticValues *element information item* has the following Infoset properties:

- 819 • A [local name] of "StaticValues" .
- 820 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 821 • zero or more *attribute information items* amongst its [attributes] as follows:

- 822 o Zero or more namespace qualified *attribute information items*. The [namespace
823 name] of such *attribute information items* MUST NOT be "http://docs.oasis-
824 open.org/wsrf/rmd-1".
- 825 • Zero or more *element information items* amongst its [children], either:
- 826 o An OPTIONAL documentation *element information item* (See section 9).
- 827 o Zero or more namespace-qualified *element information items*. The [namespace
828 name] of such *element information items* MUST NOT be "http://docs.oasis-
829 open.org/wsrf/rmd-1".
- 830 ▪ Each *element information item* MUST be an XML fragment that
831 corresponds to the type of the XML element identified by the [parent]
832 Property component's {name}
- 833 ▪ Note, because these are XML fragments, it is not expected that a
834 processor of a MetadataDescriptor document would need to validate
835 these element information items (processContents = "skip").
836

837 **8.5 InitialValues**

838 The purpose of the InitialValues component is to define the set of values that a [parent] Property
839 component's {name} will contain when a WS-Resource becomes available for the first time. If the
840 {initialValues} of a Property component is not empty, any Web service that implements the
841 portType or interface identified by {interface} MUST ensure that all the value(s) defined in
842 {initialValues} appear in the {name} when the service comes online. There is no guarantee as to
843 how long these values will be present before they are modified; they are different from values
844 defined in {staticValues} because they are mutable.

845

846 The values contained in a InitialValues component MUST conform to the XML Schema definition
847 of the {name}. Note: because the child *element information items* of a InitialValues *element*
848 *information item* are XML fragments, it is not required that these fragments be validated
849 (processContents is "skip").

850

851 The properties of a InitialValues component are as follows:

- 852 • {values} zero or more XML fragments that correspond to the type of the [parent] Property
853 component's {name}.
- 854 • The number of XML fragments within {values} MUST NOT be greater than the
855 maxOccurs facet of the schema declaration target of the {name}.

856

857 The following is an XML representation of the InitialValues component:

858

859

860
861
862
863
864

865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884

```
<InitialValues
  {anyAttribute}* >
  <documentation />?
  {any}*
</ InitialValues>
```

The InitialValues *element information item* has the following Infoset properties:

- A [local name] of "InitialValues" .
- A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- zero or more *attribute information items* amongst its [attributes] as follows:
 - Zero or more namespace qualified *attribute information items*. The [namespace name] of such *attribute information items* MUST NOT be "http://docs.oasis-open.org/wsrf/rmd-1".
- Zero or more *element information items* amongst its [children], either:
 - An OPTIONAL documentation *element information item* (See section 9).
 - Zero or more namespace-qualified *element information items*. The [namespace name] of such *element information items* MUST NOT be "http://docs.oasis-open.org/wsrf/rmd-1".
 - Each *element information item* MUST be an XML fragment that corresponds to the type of the XML element identified by the [parent] Property component's {name}
 - Note, because these are XML fragments, it is not expected that a processor of a MetadataDescriptor document would need to validate these element information items (processContents = "skip").

885 **9 Documentation Component**

886 The WS-Resource Metadata MetadataDescriptor specification uses the documentation *element*
887 *information item* as a container for human readable and/or machine processable documentation
888 in a fashion similar to that defined for WSDL 2.0 [WSDL 2.0]. The content of the *element*
889 *information item* is "mixed" content as defined in XML Schema [XML Schema]. The
890 documentation *element information item* may be contained by any *element information item*
891 defined in this specification.

892

893 The following is an XML representation of the Documentation component:

894

```
895 <documentation {anyAttribute}* >  
896   {any} *  
897 </documentation>
```

898

899 The documentation *element information item* contains the following:

900

- 901 • A [local name] of "documentation".
- 902 • A [namespace name] of "http://docs.oasis-open.org/wsrf/rmd-1".
- 903 • Zero or more attribute information items.
 - 904 ○ Zero or more namespace qualified *attribute information items*. The [namespace
905 name] of such *attribute information items* MUST NOT be "http://docs.oasis-
906 open.org/wsrf/rmd-1".
- 907 • Zero or more child *element information items* amongst its [children].

908

908 **10 Obtaining a MetadataDescriptor Document**

909 There are two mechanisms that a requestor can use to obtain a WS-Resource
910 MetadataDescriptor document:

911

- 912 1. A specific attribute extension to WSDL 1.1 portType definition
- 913 2. A specific Resource Property element.

914

914 **10.1 Extending WSDL 1.1 PortType**

915 A WS-Resource MetadataDescriptor document is associated with a WSDL 1.1 portType definition
916 using an extension of the WSDL 1.1 portType element information item. If any aspect of the
917 portType is associated with a MetadataDescriptor document, then the portType element MUST
918 be extended in the manner described below. This extension is described as follows:

919

920

921

922

```
<wsdl:definitions ...>
  <wsdl:portType ...
    wsrm:Descriptor="xs:QName"?
    wsrm:DescriptorLocation="xs:anyURI"?
  ... >
...
</wsdl:portType>
```

929

930 This definition is further constrained as follows:

931 /wsdl:portType/@wsrm:Descriptor

932 If this attribute appears on a WSDL 1.1 portType element its value MUST be a QName
933 that corresponds to a MetadataDescriptor component. Further, the value of the
934 MetadataDescriptor component contained in that document MUST have {interface} that
935 matches the QName of the portType containing @wsrm:Descriptor. Any service that
936 implements this portType MUST be associated with a MetadataDescriptor that is
937 identified by the value of this attribute.

938 /wsdl:portType/@wsrm:DescriptorLocation

939 If this attribute appears on a WSDL 1.1 portType element its value MUST be a URI. The
940 URI corresponds to a URL at which can be found more information about that
941 MetadataDescriptor namespace, such as an XML document containing a Definitions
942 element as its root element.

943

944 **10.2 Using Resoure Property Elements to expose** 945 **MetadataDescriptors**

946 Clients may find and read the wsrm:MetadataDescriptor of a WS-resource using a "metadata
947 WS-resource" that is associated with the resource. The purpose of the metadata WS-resource is
948 to expose a metadata document via its resource properties document. The endpoint reference of
949 the metadata WS-resource is of type wsrm:MetadataDescriptorReference and is exposed in the
950 original WS-resource's resource property document; the form of this resource property is:

951

952

```
<xsd:element name="MetadataDescriptorReference"
  type="wsrm:MetadataDescriptorReferenceType"/>
```

954

955 The constraints on this element are as follows:

956

957 /wsrmd:MetadataDescriptorReference

958 This element is an wsa:EndpointReference to a "metadata WS-Resource" associated
959 with the target WS-Resource. This metadata WS-Resource has a resource properties
960 document that is equivalent in content to the metadata descriptor document of the target
961 WS-Resource. The metadata WS-Resource MUST restrict its resource properties
962 document such that the cardinality of the wsrmd:MetadataDescriptor child elements is
963 one rather than zero-to-many. This allows for a single wsrmd:MetadataDescriptor to
964 describe each WS-Resource instance.

965

966

967

968 11 References

969 11.1 Normative

- 970 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
971 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 972 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI):
973 Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox
974 Corporation, August 1998.
- 975 **[WS-Addressing]** <http://www.w3.org/2005/08/addressing.pdf>
- 976 **[WS-BaseNotification]** [http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-
977 spec-pr-02.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf)
- 978 **[WS-Resource]** [http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-pr-
979 02.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-pr-02.pdf)
- 980 **[WS-ResourceProperties]** [http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-
981 spec-pr-02.pdf](http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-pr-02.pdf)
- 982 **[XML-Infoset]** *W3C Recommendation "XML Information Set". Available at*
983 <http://www.w3.org/TR/xml-infoset/>
- 984 **[XML-Names]** *W3C Recommendation "Namespaces in XML". Available at*
985 <http://www.w3.org/TR/REC-xml-names/>
986

987 11.2 Non-Normative

- 988 **[WSDL2.0]** W3C Recommendation "Web Services Description Language" Available at
989 <http://www.w3.org/TR/wsdl20/>
- 990
- 991 **[AppNotes]** [http://www.oasis-
992 open.org/apps/org/workgroup/wsrf/download.php/16355/wsrf-
993 application_notes-1.2-notes-pr-02.pdf](http://www.oasis-open.org/apps/org/workgroup/wsrf/download.php/16355/wsrf-application_notes-1.2-notes-pr-02.pdf)
994

995 **Appendix A. Acknowledgments**

996 The following individuals were members of the committee during the development of this
997 specification:

998

999 In addition, the following people made contributions to this specification:

1000

1001

1002 **Appendix B. XML Schema for WS-**
1003 **ResourceMetadataDescriptor**

1004 The XML types and elements used in this specification are defined in the following XML Schema.
1005

```
1006 <?xml version="1.0" encoding="UTF-8"?>  
1007 <!--  
1008  
1009  
1010 OASIS takes no position regarding the validity or scope of any  
1011 intellectual property or other rights that might be claimed to pertain  
1012 to the implementation or use of the technology described in this  
1013 document or the extent to which any license under such rights might or  
1014 might not be available; neither does it represent that it has made any  
1015 effort to identify any such rights. Information on OASIS's procedures  
1016 with respect to rights in OASIS specifications can be found at the  
1017 OASIS website. Copies of claims of rights made available for  
1018 publication and any assurances of licenses to be made available, or the  
1019 result of an attempt made to obtain a general license or permission for  
1020 the use of such proprietary rights by implementors or users of this  
1021 specification, can be obtained from the OASIS Executive Director.  
1022  
1023 OASIS invites any interested party to bring to its attention any  
1024 copyrights, patents or patent applications, or other proprietary rights  
1025 which may cover technology that may be required to implement this  
1026 specification. Please address the information to the OASIS Executive  
1027 Director.  
1028  
1029 Copyright (C) OASIS Open (2005-2006). All Rights Reserved.  
1030
```

1031 This document and translations of it may be copied and furnished to
1032 others, and derivative works that comment on or otherwise explain it or
1033 assist in its implementation may be prepared, copied, published and
1034 distributed, in whole or in part, without restriction of any kind,
1035 provided that the above copyright notice and this paragraph are
1036 included on all such copies and derivative works. However, this
1037 document itself may not be modified in any way, such as by removing the
1038 copyright notice or references to OASIS, except as needed for the
1039 purpose of developing OASIS specifications, in which case the
1040 procedures for copyrights defined in the OASIS Intellectual Property
1041 Rights document must be followed, or as required to translate it into
1042 languages other than English.
1043
1044 The limited permissions granted above are perpetual and will not be
1045 revoked by OASIS or its successors or assigns.
1046
1047 This document and the information contained herein is provided on an
1048 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1049 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
1050 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1051 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
1052
1053
1054 -->
1055
1056 <xsd:schema
1057 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1058 xmlns="http://www.w3.org/2001/XMLSchema"
1059 xmlns:wsa="http://www.w3.org/2005/08/addressing"
1060 xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/rp-2"
1061 targetNamespace="http://docs.oasis-open.org/wsrf/rmd-1"
1062 xmlns:wsrmd="http://docs.oasis-open.org/wsrf/rmd-1"
1063 elementFormDefault="qualified">
1064
1065 <xsd:import
1066 namespace="http://docs.oasis-open.org/wsrf/rp-2"
1067 schemaLocation="http://docs.oasis-open.org/wsrf/rp-2.xsd" />
1068
1069 <xsd:import
1070 namespace="http://www.w3.org/2005/08/addressing"
1071 schemaLocation="http://www.w3.org/2005/08/addressing.xsd" />
1072
1073

```

1074
1075 <!-- ===== Utility Types ===== -->
1076 <xsd:simpleType name="PairsOfURIType">
1077   <xsd:list itemType="xsd:anyURI" />
1078 </xsd:simpleType>
1079
1080 <!-- ===== PortType Attribute Extensions ===== -
1081 ->
1082   <xsd:attribute name="Descriptor" type="xsd:QName" />
1083
1084   <xsd:attribute name="DescriptorLocation" type="xsd:anyURI" />
1085
1086 <!-- ===== Documentation Component ===== -->
1087 <xsd:complexType name="DocumentationType" mixed="true" >
1088   <xsd:sequence>
1089     <xsd:any namespace="##any"
1090       minOccurs="0" maxOccurs="unbounded"
1091       processContents="lax" />
1092   </xsd:sequence>
1093   <xsd:anyAttribute/>
1094 </xsd:complexType>
1095
1096 <xsd:complexType name="DocumentedType">
1097   <xsd:sequence>
1098     <xsd:element name="documentation" type="wsrmd:DocumentationType"
1099       minOccurs="0" maxOccurs="1" />
1100   </xsd:sequence>
1101 </xsd:complexType>
1102
1103 <!-- ===== Definitions Component ===== -->
1104 <!--
1105 <Definitions
1106   targetNamespace="xsd:anyURI"
1107   {anyAttribute}* >
1108
1109   <documentation />?
1110   <MetadataDescriptor /> *
1111   {any}*
1112
1113 </Definitions>
1114 -->
1115
1116 <xsd:complexType name="DefinitionsType" >

```

```

1117     <xsd:complexContent>
1118         <xsd:extension base="wsrmd:DocumentedType">
1119             <xsd:sequence>
1120                 <xsd:element ref="wsrmd:MetadataDescriptor"
1121                     minOccurs="0" maxOccurs="unbounded" />
1122                 <xsd:any namespace="##other"
1123                     minOccurs="0" maxOccurs="unbounded"
1124                     processContents="lax" />
1125             </xsd:sequence>
1126             <xsd:attribute name="targetNamespace"
1127                 type="xsd:anyURI" use="required"/>
1128             <xsd:anyAttribute namespace="##other" processContents="lax"/>
1129         </xsd:extension>
1130     </xsd:complexContent>
1131 </xsd:complexType>
1132
1133 <xsd:element name="Definitions" type="wsrmd:DefinitionsType" >
1134     <xsd:key name="MetadataDescriptor">
1135         <xsd:annotation>
1136             <xsd:documentation>
1137                 To form a QName, the name of any MetadataDescriptor must be
1138                 unique within a Definitions element.
1139             </xsd:documentation>
1140         </xsd:annotation>
1141         <xsd:selector xpath="wsrmd:MetadataDescriptor" />
1142         <xsd:field xpath="@name" />
1143     </xsd:key>
1144 </xsd:element>
1145
1146 <!-- ===== MetadataDescriptor Component ===== -
1147 ->
1148 <!--
1149 <MetadataDescriptor
1150     name="xsd:NCName"
1151     interface="xsd:QName"
1152     wsdlLocation="list of xsd:anyUri"?
1153     {anyAttribute}* >
1154
1155     <documentation />?
1156     <Property /> *
1157     {any}*
1158
1159 </MetadataDescriptor>

```

```

1160 -->
1161
1162 <xsd:complexType name= "MetadataDescriptorType" >
1163   <xsd:complexContent>
1164     <xsd:extension base="wsrmd:DocumentedType">
1165       <xsd:sequence>
1166         <xsd:element ref="wsrmd:Property"
1167           minOccurs="0" maxOccurs="unbounded" />
1168         <xsd:any namespace="##other"
1169           minOccurs="0" maxOccurs="unbounded"
1170           processContents="lax" />
1171       </xsd:sequence>
1172       <xsd:attribute name="name"
1173         type="xsd:NCName" use="required"/>
1174       <xsd:attribute name="interface"
1175         type="xsd:QName" use="required"/>
1176       <xsd:attribute name="wsdlLocation"
1177         type="wsrmd:PairsOfURIType" />
1178       <xsd:anyAttribute namespace="##other" processContents="lax"/>
1179     </xsd:extension>
1180   </xsd:complexContent>
1181 </xsd:complexType>
1182
1183 <xsd:element name="MetadataDescriptor"
1184   type="wsrmd:MetadataDescriptorType" />
1185
1186 <!-- ===== Property Component ===== -->
1187 <!--
1188 <Property
1189   name="xsd:QName"
1190   mutability="[constant|appendable|mutable]" ?
1191   modifiability="[read-only|read-write]" ?
1192   subscribability="xs:boolean" ?
1193   {anyAttribute}* >
1194
1195 <documentation />?
1196 [ <ValidValues> {any}* </ValidValues> |
1197   <ValidValueRange lowerBound='xsd:simpleType'
1198     upperBound='xsd:simpleType'>
1199   </ValidValueRange> ] ?
1200 <StaticValues> {any}* </StaticValues> ?
1201
1202 {any} *

```



```

1203
1204 </Property>
1205 -->
1206 <xsd:complexType name= "PropertyType" >
1207   <xsd:complexContent>
1208     <xsd:extension base="wsrmd:DocumentedType">
1209       <xsd:sequence>
1210         <xsd:choice>
1211           <xsd:element ref="wsrmd:ValidValues"
1212             minOccurs="0" maxOccurs="1" />
1213           <xsd:element ref="wsrmd:ValidValueRange"
1214             minOccurs="0" maxOccurs="1" />
1215         </xsd:choice>
1216         <xsd:element ref="wsrmd:StaticValues"
1217           minOccurs="0" maxOccurs="1" />
1218         <xsd:any namespace="##other"
1219           minOccurs="0" maxOccurs="unbounded"
1220           processContents="lax" />
1221       </xsd:sequence>
1222       <xsd:attribute name="name"
1223         type="xsd:QName" use="required"/>
1224       <xsd:attribute name="mutability"
1225         type="wsrmd:MutabilityType" />
1226       <xsd:attribute name="modifiability"
1227         type="wsrmd:ModifiabilityType" />
1228       <xsd:attribute name="subscribability" type="xsd:boolean"
1229         default="false" />
1230       <xsd:anyAttribute namespace="##other" processContents="lax"/>
1231     </xsd:extension>
1232   </xsd:complexContent>
1233 </xsd:complexType>
1234
1235 <xsd:element name="Property" type="wsrmd:PropertyType" />
1236
1237 <xsd:simpleType name="MutabilityType">
1238   <xsd:restriction base="xsd:string" >
1239     <xsd:enumeration value="constant" />
1240     <xsd:enumeration value="appendable" />
1241     <xsd:enumeration value="mutable" />
1242   </xsd:restriction>
1243 </xsd:simpleType>
1244
1245 <xsd:simpleType name="ModifiabilityType">

```

```

1246     <xsd:restriction base="xsd:string" >
1247         <xsd:enumeration value="read-only" />
1248         <xsd:enumeration value="read-write" />
1249     </xsd:restriction>
1250 </xsd:simpleType>
1251
1252 <!-- ===== Valid Values Component ===== -->
1253 <!--
1254 <ValidValues
1255     {anyAttribute}* >
1256     <documentation />?
1257     {any}*
1258 </ValidValues>
1259 -->
1260 <xsd:complexType name= "ValidValueType" mixed="true">
1261     <xsd:sequence>
1262         <xsd:element name="documentation" type="wsrmd:DocumentationType"
1263             minOccurs="0" maxOccurs="1" />
1264
1265         <xsd:any namespace="##other"
1266             minOccurs="0" maxOccurs="unbounded"
1267             processContents="lax" />
1268     </xsd:sequence>
1269     <xsd:anyAttribute namespace="##other" processContents="lax"/>
1270 </xsd:complexType>
1271
1272 <xsd:element name="ValidValues" type="wsrmd:ValidValueType" />
1273
1274 <!-- ===== Valid Range Component ===== -->
1275 <!--
1276 <ValidValueRange
1277     lowerBound="xs:anySimpleType" ? upperBound="xs:anySimpleType" ?
1278     {anyAttribute}* >
1279     <documentation />?
1280     {any}*
1281 </ValidValueRange>
1282 -->
1283 <xsd:complexType name= "ValidValueRangeType" mixed="true">
1284     <xsd:sequence>
1285         <xsd:element name="documentation" type="wsrmd:DocumentationType"
1286             minOccurs="0" maxOccurs="1" />
1287
1288         <xsd:any namespace="##other"

```

```

1289         minOccurs="0" maxOccurs="unbounded"
1290         processContents="lax" />
1291     </xsd:sequence>
1292     <xsd:attribute name="lowerBound" type="xsd:anySimpleType" />
1293     <xsd:attribute name="upperBound" type="xsd:anySimpleType" />
1294     <xsd:anyAttribute namespace="##other" processContents="lax"/>
1295 </xsd:complexType>
1296
1297     <xsd:element name="ValidValueRange" type="wsrmd:ValidValueRangeType"
1298 />
1299
1300 <!-- ===== Static Values Component ===== -->
1301 <!--
1302 <StaticValues
1303     {anyAttribute}* >
1304     <documentation />?
1305     {any}*
1306 </StaticValues>
1307 -->
1308     <xsd:complexType name="StaticValuesType" mixed="true">
1309         <xsd:sequence>
1310             <xsd:element name="documentation" type="wsrmd:DocumentationType"
1311                 minOccurs="0" maxOccurs="1" />
1312
1313             <xsd:any namespace="##other"
1314                 minOccurs="0" maxOccurs="unbounded"
1315                 processContents="lax" />
1316         </xsd:sequence>
1317         <xsd:anyAttribute namespace="##other" processContents="lax"/>
1318     </xsd:complexType>
1319
1320     <xsd:element name="StaticValues" type="wsrmd:StaticValuesType" />
1321
1322 <!-- ===== Initial Values Component ===== -->
1323 <!--
1324 <InitialValues
1325     {anyAttribute}* >
1326     <documentation />?
1327     {any}*
1328 </InitialValues>
1329 -->
1330     <xsd:complexType name="InitialValuesType" mixed="true">
1331         <xsd:sequence>

```

```

1332     <xsd:element name="documentation" type="wsrmd:DocumentationType"
1333               minOccurs="0" maxOccurs="1" />
1334
1335     <xsd:any namespace="##other"
1336           minOccurs="0" maxOccurs="unbounded"
1337           processContents="lax" />
1338   </xsd:sequence>
1339   <xsd:anyAttribute namespace="##other" processContents="lax"/>
1340 </xsd:complexType>
1341
1342 <xsd:element name="InitialValues" type="wsrmd:InitialValuesType" />
1343
1344
1345 <!-- ===== MetadataDescriptorReference RP GED ===== -->
1346 <xsd:complexType name="MetadataDescriptorReferenceType">
1347   <xsd:complexContent>
1348     <xsd:extension base="wsa:EndpointReferenceType"/>
1349   </xsd:complexContent>
1350 </xsd:complexType>
1351
1352 <xsd:element name="MetadataDescriptorReference"
1353           type="wsrmd:MetadataDescriptorReferenceType" />
1354
1355 <!--
1356
1357 Metadata Resource RP Doc
1358
1359 This defines one property - MetadataDescriptor - which must have a
1360 cardinality of one.
1361
1362 -->
1363
1364 <xsd:element name="MetadataResourceRP" type="wsrmd:DefinitionsType"/>
1365
1366 </xsd:schema>

```

Appendix C. Revision History

Rev	Date	By Whom	What
wd-01	2004-10-07	Tom Maguire	Initial version created based on work in response to issue 10.
wd-04	2005-10-31	Dan Jemiolo	Updates to original based on TC revisions in summer/fall of 2005.
wd-06	2005-12-12	Dan Jemiolo	Clean up remaining revisions for public review. Re-inserted some features based on requests from WSDM TC.
wd-09	2006-06-04	Dan Jemiolo	Made changes related to MetadataDescriptorReference (an EPR exposed via WSRP that allows a client to read the MDD). Also (re-)added the InitialValues concept to Property.
wd-10	2006-06-19	Dan Jemiolo	Clarified nature of MetadataResourceRP and fixed some example text (T. Banks).
cd-01	2006-06-28	Dan Jemiolo	Changed status to PR.

1369 **Appendix D. Notices**

1370 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1371 that might be claimed to pertain to the implementation or use of the technology described in this
1372 document or the extent to which any license under such rights might or might not be available;
1373 neither does it represent that it has made any effort to identify any such rights. Information on
1374 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1375 website. Copies of claims of rights made available for publication and any assurances of licenses
1376 to be made available, or the result of an attempt made to obtain a general license or permission
1377 for the use of such proprietary rights by implementors or users of this specification, can be
1378 obtained from the OASIS Executive Director.

1379

1380 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1381 applications, or other proprietary rights which may cover technology that may be required to
1382 implement this specification. Please address the information to the OASIS Executive Director.

1383

1384 Copyright (C) OASIS Open (2005). All Rights Reserved.

1385

1386 This document and translations of it may be copied and furnished to others, and derivative works
1387 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1388 published and distributed, in whole or in part, without restriction of any kind, provided that the
1389 above copyright notice and this paragraph are included on all such copies and derivative works.
1390 However, this document itself may not be modified in any way, such as by removing the copyright
1391 notice or references to OASIS, except as needed for the purpose of developing OASIS
1392 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1393 Property Rights document must be followed, or as required to translate it into languages other
1394 than English.

1395

1396 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1397 successors or assigns.

1398

1399 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1400 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1401 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
1402 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1403 PARTICULAR PURPOSE.