



Web Services Resource Properties 1.2 (WS-ResourceProperties)

Working Draft 04, 10 June 2004

Document identifier:

wsrf-WS-ResourceProperties-1.2-draft-04

Location:

<http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-04.pdf>

Editors:

Steve Graham, IBM <sggraham@us.ibm.com>

Jem Treadwell, Hewlett-Packard Company <Jem.Treadwell@hp.com>

Abstract:

The relationship between Web services and stateful resources is defined in [State Paper]. This relationship is described as the implied resource pattern. In the implied resource pattern, messages to a Web service may include a component that identifies a stateful resource to be used in the execution of the message. We refer to the composition of a stateful resource and a Web service under the implied resource pattern as a WS-Resource.

This document standardizes the means by which the definition of the properties of a WS-Resource may be declared as part of a Web service interface. The declaration of the WS-Resource's properties represents a projection of or a view on the WS-Resource's state. This projection is defined in terms of a resource properties document. This resource properties document serves to define a basis for access to the resource properties through Web service interfaces.

This specification also defines a standard set of message exchanges that allow a requestor to query or update the property values of the WS-Resource. The set of properties defined in the resource properties document associated with the service interface defines the constraints on the valid contents of these message exchanges.

Status:

This document and associated schema are published by this TC as "working drafts" and represent the starting point for our standardization process. It is possible that they may change significantly during this process, but should nonetheless provide a stable reference for discussion and early adopters' implementations.

Committee members should send comments on this specification to the wsrf@lists.oasis-open.org list. Others should subscribe to and send comments to the wsrf-

36 comment@lists.oasis-open.org list. To subscribe, send an email message to [wsrf-](mailto:wsrf-comment-request@lists.oasis-open.org)
37 [comment-request@lists.oasis-open.org](mailto:wsrf-comment-request@lists.oasis-open.org) with the word "subscribe" as the body of the
38 message.

39 For information on whether any patents have been disclosed that may be essential to
40 implementing this specification, and any offers of patent licensing terms, please refer to
41 the Intellectual Property Rights section of the WSRF TC web page ([http://www.oasis-](http://www.oasis-open.org/committees/wsrf/)
42 [open.org/committees/wsrf/](http://www.oasis-open.org/committees/wsrf/)).

43 Table of Contents

44	1	Introduction.....	4
45	1.1	Goals and Requirements	4
46	1.1.1	Requirements	4
47	1.1.2	Non-Goals	5
48	1.2	Terminology	5
49	1.3	Namespaces.....	5
50	2	Terminology and Concepts.....	7
51	3	Example	9
52	4	Declaring Resource Properties.....	11
53	4.1	Resource Properties Document	11
54	4.2	Resource Properties Document Type	11
55	4.3	Declaring the Resource Properties Document Type in WSDL	12
56	4.4	Resource Properties and Interface Aggregation	12
57	5	Operations on Resource Properties.....	15
58	5.1	GetResourceProperty	15
59	5.1.1	Example SOAP Encoding of the GetResourceProperty Message Exchange.....	16
60	5.2	GetMultipleResourceProperties	17
61	5.2.1	Example SOAP Encoding of the GetMultipleResourceProperties Message	
62		Exchange	19
63	5.3	SetResourceProperties.....	20
64	5.3.1	Example SOAP Encoding of the SetResourceProperties Message Exchange	23
65	5.4	QueryResourceProperties.....	25
66	5.4.1	Example SOAP Encoding of the QueryResourceProperties Message Exchange..	26
67	6	Subscription.....	29
68	7	ACID Properties of Operations on WS-Resources	32
69	8	Security Considerations.....	33
70	8.1	Securing the message exchanges	33
71	8.2	Securing Resource Properties	34
72	9	References.....	36
73	9.1	Normative	36
74	9.2	Non-Normative.....	36
75		Appendix A. Acknowledgments.....	37
76		Appendix B. XML Schema	38
77		Appendix C. WSDL 1.1	41
78		Appendix D. Revision History.....	51
79		Appendix E. Notices.....	52
80			

81 1 Introduction

82 The relationship between Web services and stateful resources is defined in [State Paper]. This
83 relationship is described as the *implied resource pattern*. In the implied resource pattern,
84 messages to a Web service include a component that identifies a stateful resource to be used in
85 the execution of the message exchange. We refer to the composition of a stateful resource and a
86 Web service under the implied resource pattern as a WS-Resource.

87 This specification standardizes the means by which the definition of the properties of a WS-
88 Resource may be declared as part of the Web service interface. The declaration of the WS-
89 Resource's properties represents a projection of or a *view* on the WS-Resource's state. The
90 projection is defined in terms of a resource properties document. This resource properties
91 document serves to define a basis for access to the resource properties through the Web service
92 interface.

93 This specification also defines a standard set of message exchanges that allow a requestor to
94 query or update the property values of the implied resource. The set of properties defined in the
95 resource properties document, and associated with the service interface, defines the constraints
96 on the valid contents of these message exchanges.

97 In this document, we outline the goals and requirements for resource properties. We define the
98 means to declare resource properties as part of a Web service description. Following this, we
99 define the message exchanges for querying and updating resource property values. The
100 document concludes with a discussion of security considerations, including a discussion of
101 security considerations associated with resource properties. As an appendix, we provide
102 normative XML and WSDL descriptions of resource properties.

103 WS-ResourceProperties is inspired by a portion of the Global Grid Forum's "Open Grid
104 Services Infrastructure (OGSI) Version 1.0" specification [OGSI].

105 1.1 Goals and Requirements

106 The goal of WS-ResourceProperties is to standardize the terminology, concepts, operations,
107 WSDL and XML needed to express the resource properties projection, its association with the
108 Web service interface, and the messages defining the query and update capability against the
109 properties of a WS-Resource.

110 1.1.1 Requirements

111 In meeting this goal, the specification must address the following specific requirements:

112 **This specification MUST:**

- 113 • Define the term "resource property" and its relationship to Web services and WS-Resources.
- 114 • Define the means by which a designer decorates a Web service description with the names
115 and types of properties associated with a WS-Resource.
- 116 • Define the means by which a requestor can:
 - 117 • Retrieve the values of one or more properties of a WS-Resource
 - 118 • Update the values of one or more properties of a WS-Resource
 - 119 • Query across the values of one or more properties of a WS-Resource
 - 120 • Subscribe for notification [WS-Notification] when the value of a WS-Resource property
121 changes.

122 The means by which resource property values are retrieved and updated SHOULD reflect a
123 document-oriented style and MUST provide the means to perform batched query and update
124 against the implied resource in a single message exchange. This will facilitate improved
125 performance over approaches requiring a separate request message exchange for each
126 individual resource property access.

127 Web services are often described using a collection of message exchange sets (e.g. WSDL 1.1
128 portTypes). These message exchange sets may be aggregated (using manual cut-and-paste in
129 WSDL 1.1) to form the "final" composed interface definition for the Web service. The requestor's
130 exposure to and interpretation of the Web service interface may be defined by a partial subset of
131 the constituent message exchange sets in the overall interface composition. Therefore, a
132 requestor will form resource property-related message requests based on this potentially partial
133 understanding of the overall composed interface to the Web service. It MUST be possible for a
134 requestor, having partial knowledge of the composed service interface, to form correct and
135 consistent resource property access message requests that execute properly on a Web service
136 that implements an extended message exchange set.

137 1.1.2 Non-Goals

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

139 General purpose XML document query and update: This specification is not meant to be used for
140 querying and updating generic XML documents, or to be used outside the context of modeling
141 stateful resources with Web services.

142 1.2 Terminology

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

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

149 When describing concrete XML schemas, this specification uses the notational convention of
150 [WS-Security]. Specifically, each member of an element's [children] or [attributes] property is
151 described using an XPath-like notation (e.g., /x:MyHeader/x:SomeProperty/@value1). The use of
152 {any} indicates the presence of an element wildcard (<xsd:any/>). The use of @{any} indicates
153 the presence of an attribute wildcard (<xsd:anyAttribute/>).

154 1.3 Namespaces

155 The following namespaces are used in this document:

Prefix	Namespace
s12	http://www.w3.org/2003/05/soap-envelope
xsd	http://www.w3.org/2001/XMLSchema
wsp	http://schemas.xmlsoap.org/ws/2002/12/policy
wsa	http://schemas.xmlsoap.org/ws/2003/02/addressing

wsrp	http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-01.xsd
wsrpw	http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-01.wsdl
wsbf	http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-BaseFaults-1.2-draft-01.xsd
wsbfw	http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-BaseFaults-1.2-draft-01.wsdl
wsnt	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd
wsntw	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl

156

2 Terminology and Concepts

157

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

160 **WS-Resource:**

- 161 o A Web service having an association with a stateful resource, where the stateful resource
162 is defined by a resource properties document type and the association is expressed by
163 annotating a WSDL portType with the type definition of the resource properties
164 document.

165 **Implied Resource Pattern:**

- 166 o The way WS-Addressing must be used to designate the stateful resource component of
167 the WS-Resource to be used in the execution of message exchanges.
- 168 o An EndpointReference that follows the implied resource pattern may include a
169 ReferenceProperties child element that identifies the stateful resource component of the
170 WS-Resource to be used in the execution of all message exchanges performed using
171 this EndpointReference.
- 172 o A message that follows the implied resource pattern **MUST** be sent to a Web service
173 referred to by an EndpointReference that follows the implied resource pattern, and **MUST**
174 conform to the WS-Addressing requirements on that message including adding the
175 ReferenceProperties information, if present, from that EndpointReference to the
176 message.
- 177 o A Web service that follows the implied resource pattern **MAY** use the
178 ReferenceProperties information from a message that follows the implied resource
179 pattern in order to identify the stateful resource to be used in the execution requested by
180 that message.

181 **WS-Resource Qualified Endpoint Reference:**

- 182 o An Endpoint Reference used to refer to a WS-Resource composed of a Web service and
183 a stateful resource.
- 184 o A stateful resource identifier **MAY** be contained within the ReferenceProperties element
185 of the Endpoint Reference.
- 186 o The address of the Web service associated with the WS-Resource **MUST** be contained in
187 the Address element of the Endpoint Reference.

188 **Resource Property:**

- 189 o A resource property is a piece of information defined as part of the state model of a WS-
190 Resource.
- 191 o A resource property may reflect a part of the resource's state, meta-data, manageability
192 information, etc.

193 **Resource Properties Document:**

- 194 o The XML document representing a logical composition of resource property elements.
195 The resource properties document defines a particular view or projection of the state data
196 implemented by the WS-Resource.
- 197 o The type (e.g. the XML Schema definition of the root element) of a resource properties
198 document is associated with the WSDL portType defining the Web service interface. This
199 association is the basis of the WS-Resource definition. Each instance of a particular WS-

200 Resource type MUST implement a logical resource properties document of the type
201 declared in the WSDL portType.

202 **Resource Property Element:**

- 203 ○ The XML representation of a resource property.
- 204 ○ A resource property element must appear as the immediate child of the root element of a
205 resource properties document.
- 206 ○ A resource property element must be an XML global element definition (GED), and is
207 uniquely identified by QName.

208 **Resource Property Value:**

- 209 ○ The value(s) associated with a resource property.

210

3 Example

211 The simple example below defines the GenericDiskDrive portType and the resource properties
212 document associated with GenericDiskDrive. The association of the resource properties
213 document with the portType defines the type of the WS-Resource.

```
214 <wsdl:definitions ... xmlns:tns="http://example.com/diskDrive" ...>
215 ...
216 <wsdl:types>
217 <xsd:schema targetNamespace="http://example.com/diskDrive" ... >
218
219 <!-- Resource property element declarations -->
220 <xsd:element name="NumberOfBlocks" type="xsd:integer"/>
221 <xsd:element name="BlockSize" type="xsd:integer" />
222 <xsd:element name="Manufacturer" type="xsd:string" />
223
224 <!-- Resource properties document declaration -->
225 <xsd:element name="GenericDiskDriveProperties">
226 <xsd:complexType>
227 <xsd:sequence>
228 <xsd:element ref="tns:NumberOfBlocks"/>
229 <xsd:element ref="tns:BlockSize" />
230 <xsd:element ref="tns:Manufacturer" />
231 <xsd:any minOccurs="0" maxOccurs="unbounded" />
232 </xsd:sequence>
233 </xsd:complexType>
234 </xsd:element>
235 ...
236 </xsd:schema>
237 </wsdl:types>
238 ...
239 <!-- Association of resource properties document to a portType -->
240 <wsdl:portType name="GenericDiskDrive"
241 <wsrp:ResourceProperties="tns:GenericDiskDriveProperties" >
242
243 <operation name="start" .../>
244 <operation name="stop" .../>
245 ...
246 </wsdl:portType>
247 ...
248 </wsdl:definitions>
```

249 The following represents the request message used to retrieve two resource property elements
250 from the WS-Resource that implements the GenericDiskDrive portType:

```
251 ...
252 <wsrp:GetMultipleResourceProperties
253 <xmlns:tns="http://example.com/diskdrive" ...>
254 <wsrp:ResourceProperty>tns:NumberOfBlocks</wsrp:ResourceProperty>
255 <wsrp:ResourceProperty>tns:BlockSize</wsrp:ResourceProperty>
```

256 </wsrp:GetMultipleResourceProperties>
257 ...

258 The following is a sample response to the simple get request:

259 ...
260 <wsrp:GetMultipleResourcePropertiesResponse
261 xmlns:ns1="http://example.com/diskdrive" ...>
262 <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
263 <ns1:BlockSize>1024</ns1:BlockSize>
264 </wsrp:GetMultipleResourcePropertiesResponse>
265 ...

266 4 Declaring Resource Properties

267 4.1 Resource Properties Document

268 The resource properties document type associated with a Web service's WSDL 1.1 portType
269 definition provides the declaration of the exposed resource properties of the WS-Resource. It
270 represents a particular composed structural view or projection of the resource properties of the
271 WS-Resource, essentially exposing the stateful resource component within the WS-Resource
272 composition. This may be used by a service requestor to form an XML-based query or update
273 expression on the WS-Resource.

274 This specification does not dictate the means by which a service implements a resource
275 properties document. A given service implementation may choose to realize its implementation of
276 the resource properties document as an actual XML instance document, stored in memory, in the
277 file system, in a database or in some XML Repository. Other service implementations may
278 *dynamically* construct the resource property elements and their values, from data held in
279 programming language objects (such as a J2EE EJB Entity Bean) or by executing a command on
280 a private communications channel to a physical resource. Yet another implementation possibility
281 is a mapping layer to a standard management interface (such as CIM or SNMP).

282 There is an explicit relationship between the resource properties document and the message
283 exchanges defined in Section 5. Any Web service that implements an interface that includes a
284 resource properties document type declaration is a WS-Resource. A WS-Resource **MUST** accept
285 message requests declared by the GetResourceProperty message exchange defined in Section
286 5. Similarly, such a Web service **MAY** accept message requests declared by the other message
287 exchanges defined in Section 5.

288 However, there is no relationship, intended or implied by this specification, between the resource
289 properties defined in the resource properties document and any other message exchanges that
290 may be introduced as part of the Web service interface. Any relationships between the resource
291 properties and messages that comprise an interface are entirely under the purview of the
292 designer of that interface. For example, using the resource properties document described above
293 in Section 3, it would be legal for an interface designer to introduce a "getNumberOfBlocks"
294 message exchange. However, with respect to this specification, there is no relationship either
295 required or prohibited between such an operation and the properties declared in the resource
296 properties document.

297 4.2 Resource Properties Document Type

298 A *resource properties document* **MUST** be defined using the following rules:

- 299 1. The resource properties document **MUST** be a global element declaration (GED) in some
300 XML namespace. This GED defines the type of the root element of a resource properties
301 document and hence the type of the resource properties document itself.
- 302 2. The resource properties document **MUST** be uniquely identified by a QName.
- 303 3. The complexType defining the resource properties document **MUST** define element
304 children only; it **MUST NOT** define attributes. The child elements **MUST** be aggregated
305 using xsd:sequence or xsd:all. The order of appearance of the resource properties within
306 the resource properties document does not matter to WS-ResourceProperties.
- 307 4. The complexType defining the resource properties document **MUST** define a sequence
308 of one or more child elements, called *resource property elements*.

- 309 a) Child elements MUST be defined using XML schema element reference (@ref).
310 b) This specification defines no additional restriction on the use of @minOccurs or
311 @maxOccurs or other information elements associated with the XML Schema
312 element definition.
- 313 5. The complexType defining the resource properties document MAY allow open element
314 content (xsd:any).

315 4.3 Declaring the Resource Properties Document Type in WSDL

316 The resource properties document definition is associated with a Web service WSDL 1.1
317 portType in the following manner:

```
318 <wsdl:definitions ...>  
319   <wsdl:portType ...  
320     wsrp:ResourceProperties="xsd:QName"? ... >  
321   ...  
322 </wsdl:portType>
```

323 This definition is further constrained as follows:

324 /wsdl:portType/@wsrp:ResourceProperties

325 If this attribute appears on a WSDL 1.1 portType element (using attribute extensibility
326 available in the WSDL 1.1 XML schema definition for the portType element) its value
327 MUST be a QName referring to a resource properties document as defined in Section
328 4.2.

329 Any service that implements a portType annotated with @wsrp:ResourceProperties MUST be a
330 component of a WS-Resource and MUST provide the interface to resource properties via a
331 document whose root element is defined by the XML global element declaration associated with
332 the portType.

333 4.4 Resource Properties and Interface Aggregation

334 Web service interface designers MAY define a collection of discrete interfaces (portTypes), each
335 of which defines a set of message exchange patterns (operations). A common design scenario is
336 one in which the designer combines these discrete interfaces to form a composed, *most-derived*
337 interface of a Web service. Examples of independently-specified interfaces designed for purposes
338 of aggregation into a most-derived interface include WS-Notification [WS-Notification], WS-
339 ResourceLifetime [WS-ResourceLifetime], and a large number of general-purpose or application-
340 domain-specific management interfaces. Further, there may be various dependencies between
341 these interfaces. That is, the messages defined by interface A may only be useful in a service
342 implementation when combined with those of interface B.

343 Within WSDL 1.1, there is no formally-defined interface extension mechanism¹. In WSDL 1.1 we
344 expect service designers to *copy-and-paste* operations from the various constituent interfaces
345 into a single, flat, most-derived service interface. In addition, we expect the service interface

¹ WSDL 2.0 is expected to define a mechanism to formally model interface aggregation /interface/@extends [WSDL 2.0].

346 designer to compose a resource property document for the most-derived Web service interface
347 that consists of all of the resource property element declarations from each of the constituent
348 interfaces used in the composition.

349 Consider the following example, wherein a designer extends the "GenericDiskDrive" WS-
350 Resource interface in a vendor-specific fashion.

```
351 <wsdl:definitions ...
352     xmlns:gen="http://example.com/diskDrive"
353     xmlns:ven="http://vendor.com/diskDrive"
354     ...>
355 ...
356 <wsdl:types>
357     <xsd:schema targetNamespace="http://vendor.com/diskDrive" ... >
358
359         <!-- Resource property element declarations -->
360         <xsd:element name="VendorExtension" type="xsd:string" />
361
362         <!-- Resource properties document declaration -->
363         <xsd:element name="VendorDiskDriveProperties">
364             <xsd:complexType>
365                 <xsd:sequence>
366                     <xsd:element ref="gen:NumberOfBlocks"/>
367                     <xsd:element ref="gen:BlockSize" />
368                     <xsd:element ref="gen:Manufacturer" />
369                     <xsd:element ref="ven:VendorExtension" />
370                     <xsd:any minOccurs="0" maxOccurs="unbounded" />
371                 </xsd:sequence>
372             </xsd:complexType>
373         </xsd:element>
374 ...
375     </xsd:schema>
376 </wsdl:types>
377 ...
378 <!-- Association of resource properties document to a portType -->
379 <wsdl:portType name="VendorDiskDrive"
380     wsrp:ResourceProperties="ven:VendorDiskDriveProperties" >
381     <operation name="...
382 ...
383     <!-- copy/paste operations from genericDiskDrive -->
384     <operation name="start" .../>
385     <operation name="stop" .../>
386 ...
387     <!-- define Vendor-specific operations -->
388     <operation name="reset" .../>
389 ...
390 </wsdl:portType>
391 ...
392 </wsdl:definitions>
393
```

394 The VendorDiskDrive portType is an example of *manual* interface aggregation in WSDL 1.1 using
395 copy-and-paste. In this example, the designer of the VendorDiskDrive portType wishes to *extend*
396 the GenericDiskDrive portType.

397 WS-ResourceProperties specifies that this style of extension MUST be carried out in the following
398 fashion:

399 1. Define the new portType.

400 In this example the new portType is named "VendorDiskDrive". This portType extends
401 "GenericDiskDrive".

402 2. Copy all of the operation child elements from the portType being extended, and paste
403 them as child elements of the new portType; the order of the operations SHOULD be
404 preserved.

405 In this example, the "start" and "stop" operations are copied from the GenericDiskDrive
406 portType and pasted as child elements of the VendorDiskDrive portType.

407 3. Define additional, vendor-specific operations as child elements of the new portType.

408 In this example, the "reset" operation is a new operation defined by the VendorDiskDrive
409 portType.

410 4. Define a new resource properties document, as an XML global element declaration,
411 following the requirements defined in Section 4.2.

412 In this example, the element is named "VendorDiskDriveProperties" and defined in the
413 "http://vendor.com/diskDrive" namespace.

414 5. Copy all of the child elements (@ref and xsd:any) from the resource properties document
415 of the portType being extended, and paste them as child elements of the new resource
416 properties document; the order of the elements SHOULD be preserved. This step MUST
417 be repeated for each portType that is being extended by this new portType. Any duplicate
418 child elements MUST be removed.

419 In this example, the elements that reference (@ref) "gen:NumberOfBlocks",
420 "gen:Blocksize", and "gen:Manufacturer" and the "xsd:any" are copied from the
421 GenericDiskDriveProperties declaration and pasted to the VendorDiskDriveProperties
422 declaration.

423 6. Define any additional resource property elements that are specific to the newly-defined
424 resource properties document type.

425 In this example, VendorDiskDriveProperties resource document defines an additional
426 resource property named VendorExtension.

5 Operations on Resource Properties

427

428 This section defines a collection of message exchanges that standardize the means by which a
429 requestor can retrieve values of resource properties, update values of resource properties, and
430 issue queries against resource properties.

431 Any interface that includes a resource properties document type declaration
432 (/wsdl:portType/@ResourceProperties) MUST also include the GetResourceProperty message
433 exchange (operation) defined in this section. Any Web service that implements an interface that
434 includes a resource properties document type declaration MAY also support the other message
435 exchanges defined in this section.

5.1 GetResourceProperty

436
437 A Web service that implements a portType that includes the resource properties document type
438 declaration (/wsdl:portType/@ResourceProperties) is a component of a WS-Resource, and
439 MUST support the message exchange defined in this section that allows a requestor to retrieve
440 the value of a single resource property of a WS-Resource.

441 The format of this request message MUST be:

442

```
443 <wsrp:GetResourceProperty>  
444   QName  
445 </wsrp:GetResourceProperty>
```

446

447 The GetResourceProperty request message MUST follow the implied resource pattern, as
448 defined in Section 2.

449 The components of the GetResourceProperty request message are further described as follows:

450 /wsrp:GetResourceProperty/QName

451 This MUST correspond to the QName of a resource property element defined as a child
452 of the root of the WS-Resource's resource properties document.

453 The response of the GetResourceProperty request message is a message of the following form:

454

```
455 <wsrp:GetResourcePropertyResponse>  
456   {any} *  
457 </wsrp:GetResourcePropertyResponse>
```

458

459 The contents of the GetResourceProperty response message are further described as follows:

460 /wsrp:GetResourcePropertyResponse/{any}

461 The resource property value, as an XML element, that corresponds to the QName in the
462 GetResourceProperty request. Note: in the case where the resource property element is
463 defined with minOccurs="0" and the resource properties document does not contain any
464 value for that resource property, the response MUST be an empty
465 wsrpw:GetResourcePropertyResponse element.

466 If the WS-Resource does not respond to the GetResourceProperty request message with the
467 GetResourcePropertyResponse message, then it MUST send one of the following fault
468 messages:

- 469 • ResourceUnknownFault

470 o The resource identified in the message (which follows the implied resource pattern) is
471 not known to the Web service.

472 • InvalidResourcePropertyQName

473 o The QName in the request message did not correspond to a resource property
474 element of the WS-Resource referred to in the request message.

475 OtherFaults: tbd

476 Note: All faults generated must be compliant with the WS-BaseFaults [WS-BaseFaults]
477 specification.

478 5.1.1 Example SOAP Encoding of the GetResourceProperty Message 479 Exchange

480 Consider the following resource properties document defining resource properties for a WS-
481 Resource defined by the GenericDiskDrive portType:

```
482 <GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >  
483   <tns:NumberOfBlocks>22</tns:NumberOfBlocks>  
484   <tns:BlockSize>1024</tns:BlockSize>  
485   <tns:Manufacturer>DrivesRUs</tns:Manufacturer>  
486 </GenericDiskDriveProperties>
```

487 The following is a non-normative example of a GetResourceProperty request message using
488 SOAP 1.2 [SOAP 1.2]:

```
489 <s12:Envelope  
490   xmlns:s12="http://www.w3.org/2003/05/soap-envelope"  
491   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"  
492   xmlns:wsrp=  
493   "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-  
494   1.2-draft-01.xsd"  
495   xmlns:ex="http://example.com/exampleNS">  
496   <s12:Header>  
497     <wsa:Action>  
498       http://docs.oasis-open.org/wsrp/2004/06/WS-  
499   ResourceProperties/GetResourceProperty  
500     </wsa:Action>  
501     <wsa:To s12:mustUnderstand="1">  
502       http://www.provider.org/ProviderEndpoint  
503     </wsa:To>  
504     <ex:ResourceDisambiguator>  
505       uuid:84dec55-7d3f-65ad-ac44-675d9fce5d22  
506     </ex:ResourceDisambiguator>  
507   </s12:Header>  
508   <s12:Body>  
509     <wsrp:GetResourceProperty  
510       xmlns:tns="http://example.com/diskDrive">  
511       tns:NumberOfBlocks  
512     </wsrp: GetResourceProperty>  
513   </s12:Body>
```

514 </s12:Envelope>

515 The following is an example GetResourcePropertyResponse message using SOAP 1.2 [SOAP
516 1.2]:

```
517 <s12:Envelope
518   xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
519   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
520   xmlns:wsrp=
521     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
522 ResourceProperties-1.2-draft-01.xsd"
523   xmlns:resp="http://www.other.org/otherNS">
524   <s12:Header>
525     <wsa:Action>
526       http://docs.oasis-open.org/wsrp/2004/06/WS-
527 ResourceProperties/GetResourcePropertyResponse
528     </wsa:Action>
529     <wsa:To s12:mustUnderstand="1">
530       http://www.requestor.org/someEndpoint
531     </wsa:To>
532     <resp:SomeResourceRef>
533       uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
534     </resp:SomeResourceRef>
535   </s12:Header>
536   <s12:Body>
537     <wsrp:GetResourcePropertyResponse
538       xmlns:ns1="http://example.com/diskDrive">
539       <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
540     </wsrp:GetResourcePropertyResponse>
541   </s12:Body>
542 </s12:Envelope>
```

543 5.2 GetMultipleResourceProperties

544 A Web service that implements a portType that includes the resource properties document type
545 declaration (/wsdl:portType/@ResourceProperties) is a component of a WS-Resource, and MAY
546 support the message exchange defined in this section that allows a requestor to retrieve the
547 values of multiple resource properties of a WS-Resource.

548 The format of this request message MUST be:

```
549 <wsrp:GetMultipleResourceProperties>
550   <wsrp:ResourceProperty>QName <wsrp:ResourceProperty>+
551 </wsrp:GetMultipleResourceProperties>
```

552 The GetMultipleResourceProperties request message MUST follow the implied resource pattern,
553 as defined in Section 2.

554 The components of the GetMultipleResourceProperties request message are further described as
555 follows:

556 /wsrp:GetMultipleResourceProperties/wsrp:ResourceProperty+

557 This component MAY appear one or more times. Each ResourceProperty element
558 contains an xsd:QName which MUST correspond to the QName of a resource property
559 element defined as a child of the root of the WS-Resource's resource properties
560 document.

561 The response of the GetMultipleResourceProperties request message is a message of the
562 following form:

```
563 <wsrp:GetMultipleResourcePropertiesResponse>  
564 {any} *  
565 </wsrp:GetMultipleResourcePropertiesResponse>
```

566 The contents of the GetMultipleResourcePropertiesResponse message are further described as
567 follows:

568 /wsrp:GetMultipleResourcePropertiesResponse/{any}

569 A collection of resource property values, as XML elements that correspond to the
570 QNames given in the GetMultipleResourceProperties request message. This collection is
571 formed in the following fashion. For each QName in the request message, the resource
572 must add to the collection all child elements of the root of the resource properties
573 document whose name corresponds to that QName. Note: in the case where the
574 resource property element is defined with minOccurs="0" and the resource properties
575 document does not contain any value for that resource property, no child element is
576 added to the collection for that QName.

577 If the XML schema definition of the resource properties document root element does not
578 permit the root element to contain a child element with that QName the processing of the
579 GetMultipleResourceProperties request message MUST terminate with a fault message.

580 The collection of resource property values SHOULD be formed in the same order as the
581 resource property element QNames were specified in the GetMultipleResourceProperties
582 request message.

583 If the WS-Resource does not respond to the GetMultipleResourceProperties request message
584 with the GetMultipleResourcePropertiesResponse message, then it MUST send one of the
585 following fault messages:

- 586 • ResourceUnknownFault
 - 587 ○ The resource identified in the message (which follows the implied resource pattern) is
588 not known to the Web service.
- 589 • InvalidResourcePropertyQName
 - 590 ○ One or more of the QNames in the request message did not correspond to a
591 resource property element of the WS-Resource referred to in the request message.

592 OtherFaults: tbd

593 Note: All faults generated must be compliant with the WS-BaseFaults [WS-BaseFaults]
594 specification.

595 Note: the functionality provided by the GetResourceProperty message exchange is a strict subset
596 of that provided by GetMultipleResourceProperties. WS-ResourceProperties defines two
597 message exchange sets to provide implementation flexibility. GetResourceProperty is a simple,
598 required message exchange that allows simple Web service implementations to be compliant
599 with WS-ResourceProperties. The optional GetMultipleResourceProperties, while more

600 sophisticated, allows efficient retrieval of multiple resource property values using a single
601 message exchange.

602 An example use of the GetMultipleResourceProperties operation is shown in Section 3. Note: it is
603 the responsibility of the requestor to correlate the elements of the response message that
604 correspond to the QNames contained in the request message.

605 **5.2.1 Example SOAP Encoding of the GetMultipleResourceProperties** 606 **Message Exchange**

607 Consider the following resource properties document defining resource properties for a WS-
608 Resource defined by the GenericDiskDrive portType:

```
609 <GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >  
610   <tns:NumberOfBlocks>22</tns:NumberOfBlocks>  
611   <tns:BlockSize>1024</tns:BlockSize>  
612   <tns:Manufacturer>DrivesRUs</tns:Manufacturer>  
613 </GenericDiskDriveProperties>
```

614 The following is a non-normative example of a GetMultipleResourceProperties request message
615 using SOAP 1.2 [SOAP 1.2]:

```
616 <s12:Envelope  
617   xmlns:s12="http://www.w3.org/2003/05/soap-envelope"  
618   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"  
619   xmlns:wsrp=  
620     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-  
621 ResourceProperties-1.2-draft-01.xsd"  
622   xmlns:ex="http://example.com/exampleNS">  
623   <s12:Header>  
624     <wsa:Action>  
625       http://docs.oasis-open.org/wsrp/2004/06/WS-  
626 ResourceProperties/GetMultipleResourceProperties  
627     </wsa:Action>  
628     <wsa:To s12:mustUnderstand="1">  
629       http://www.provider.org/ProviderEndpoint  
630     </wsa:To>  
631     <ex:ResourceDisambiguator>  
632       uuid:84dec55-7d3f-65ad-ac44-675d9fce5d22  
633     </ex:ResourceDisambiguator>  
634   </s12:Header>  
635   <s12:Body>  
636     <wsrp:GetMultipleResourceProperties  
637       xmlns:tns="http://example.com/diskdrive">  
638       <wsrp:ResourceProperty>tns:NumberOfBlocks</wsrp:ResourceProperty>  
639       <wsrp:ResourceProperty>tns:BlockSize</wsrp:ResourceProperty>  
640     </wsrp:GetMultipleResourceProperties>  
641   </s12:Body>  
642 </s12:Envelope>
```

643 The following is an example GetMultipleResourcePropertiesResponse message using SOAP 1.2
644 [SOAP 1.2]:

```

645 <s12:Envelope
646     xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
647     xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
648     xmlns:wsrp=
649     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
650 ResourceProperties-1.2-draft-01.xsd"
651     xmlns:resp="http://www.other.org/otherNS">
652   <s12:Header>
653     <wsa:Action>
654       http://docs.oasis-open.org/wsrp/2004/06/WS-
655 ResourceProperties/GetMultipleResourcePropertiesResponse
656     </wsa:Action>
657     <wsa:To s12:mustUnderstand="1">
658       http://www.requestor.org/someEndpoint
659     </wsa:To>
660     <resp:SomeResourceRef>
661       uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
662     </resp:SomeResourceRef>
663   </s12:Header>
664   <s12:Body>
665     <wsrp:GetMultipleResourcePropertiesResponse
666       xmlns:ns1="http://example.com/diskdrive" ...>
667       <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
668       <ns1:BlockSize>1024</ns1:BlockSize>
669     </wsrp:GetMultipleResourcePropertiesResponse>
670   </s12:Body>
671 </s12:Envelope>

```

672 5.3 SetResourceProperties

673 A Web service that implements a portType that includes the resource properties document type
674 declaration (/wsdl:portType/@ResourceProperties) is a component of a WS-Resource, and MAY
675 support the message exchange defined in this section that allows a requestor to modify the
676 values of multiple resource properties of a WS-Resource.

677 The SetResourceProperties message allows the processing of a single request message to make
678 multiple changes to the resource properties document. There are three types of changes, each
679 modeled as separate types of component (called SetRequestComponent) of a
680 SetResourceProperties request message:

- 681 a) Insert: wherein a new resource property element is inserted into the resource properties
682 document;
- 683 b) Update: wherein existing resource property element(s) are modified; and
- 684 c) Delete: wherein existing resource property element(s) are removed.

685 The format of this request message MUST be:

```

686 <wsrp:SetResourceProperties>
687 {
688   <wsrp:Insert >
689     {any}*

```

```

690     </wsrp:Insert> |
691
692     <wsrp:Update >
693         {any}*
694     </wsrp:Update> |
695
696     <wsrp>Delete ResourceProperty="QName" />
697 }+
698 </wsrp:SetResourceProperties>

```

699 The SetResourceProperties request message MUST follow the implied resource pattern, as
700 defined in Section 2.

701 The contents of the SetResourceProperties request message are further described as follows:

702 /wsrp:SetResourceProperties

703 This element contains a collection of one or more components called
704 SetRequestComponents. Each of the SetRequestComponents must be processed
705 against the WS-Resource's resource properties document. These
706 SetRequestComponents MUST appear to be processed in the order in which they are
707 listed in the request. Each request component MUST be processed to completion in this
708 conceptual sequence before a subsequent SetRequestComponent is processed. The
709 result of processing a given SetRequestComponent MUST be observable to the
710 processing of a subsequent SetRequestComponent, and to subsequent message
711 exchanges with the same WS-Resources.

712 If a service fails to process a SetRequestComponent, it MUST cease processing the
713 SetResourceProperties request message. The values of the resource properties
714 associated with this SetRequestComponent MAY reflect partial processing of this
715 SetRequestComponent. An implementation MAY restore the contents of the resource
716 properties document to a state as if no processing of the failed SetRequestComponent
717 had occurred. The implementation MAY additionally choose to restore the resource
718 properties document as if none of the SetRequestComponents had been processed.
719 Refer to Section 7 for additional information of resource recovery.

720 /wsrp:SetResourceProperties/wsrp:Insert

721 The intent of this component is to insert the contents of the component into the resource
722 properties document. The exact placement of the element insertion is implementation-
723 dependent. If, as a result of processing the Insert component, the resource properties
724 document is no longer able to validate, the processing of the component MUST fault. The
725 implementation may be unable to accept the insertion of an element because it does not
726 allow the requestor to insert a resource property (or its value) of that given name. In such
727 circumstances, the resource MUST fault the processing of the component.

728 /wsrp:SetResourceProperties/wsrp:Insert/{any}

729 This component identifies the element(s) to be inserted into the resource properties
730 document. If there are multiple child elements of the Insert component, each MUST have
731 the same namespace and name (i.e. the same QName). The QName MUST correspond
732 to the QName of a resource property element associated with the WS-Resource (i.e. an
733 element that is a valid child element of the root element of the resource properties

734 document). Note, for those resource properties documents that allow open element
735 content, the set of valid content types can be very large.

736 /wsrp: SetResourceProperties/wsrp:Update

737 The intent of this component is to change the value of the resource property by removing
738 any and all resource property element(s) of the given QName and replacing them with
739 the contents of this component. If, as a result of processing the Update component, the
740 resource properties document is no longer able to validate, the processing of the
741 component MUST fault. The resource may be unable to accept the update of an element
742 because it does not allow the requestor to update a resource property (or its value) of
743 that given name. In such circumstances, the resource MUST fault the processing of the
744 component.

745 /wsrp: SetResourceProperties/wsrp:Update/{any}

746 This identifies the element(s) to be inserted into the resource properties document,
747 replacing all element children of the root of the resource properties document with the
748 same QName. If there are multiple child elements of the Insert component, each MUST
749 have the same namespace and name (i.e. the same QName). The QName MUST
750 correspond to the QName of a resource property element associated with the WS-
751 Resource (i.e. an element that is a valid child element of the root element of the resource
752 properties document). Note, for those resource properties documents that allow open
753 element content, the set of valid content types can be very large.

754 /wsrp: SetResourceProperties/wsrp>Delete

755 The intent of this component is to remove all element children of the root of the resource
756 properties document whose QNames correspond to the value of @ResourceProperty. If
757 the resource is unable to remove all identified elements, the processing of the component
758 MUST fault. If, as a result of processing the Delete component, the resource properties
759 document is no longer able to validate, the processing of the component MUST fail. The
760 resource may be unable to accept the delete of an element because it does not allow the
761 requestor to delete a resource property (or its value) of the given name. In such
762 circumstances, the resource MUST fault the processing of the component.

763 /wsrp: SetResourceProperties/wsrp>Delete/@ResourceProperty

764 This attribute contains the QName of a resource property to be deleted by this
765 component.

766 The response of the SetResourceProperties request message, all of whose components were
767 successfully processed, MUST be a message of the following form:

```
768 <wsrp:SetResourcePropertiesResponse>  
769 </wsrp:SetResourcePropertiesResponse>
```

770 If the WS-Resource does not respond to the SetResourceProperties request message with the
771 SetResourcePropertiesResponse message, then it MUST send one of the following fault
772 messages. For those faults associated with failure to process a SetResourceProperties request
773 component, the offending component MUST be identified in the fault message:

- 774 • ResourceUnknownFault
 - 775 ○ The resource identified in the message is not known to the Web service.
- 776 • InvalidSetResourcePropertiesRequestContent:

- 777 ○ The contents of the SetResourceProperties request component causes the resource
 - 778 ○ properties document to no longer validate.
 - 779 • UnableToModifyResourceProperty:
 - 780 ○ A resource property identified by one of the SetResourceProperties request
 - 781 ○ components is read-only.
 - 782 • InvalidResourcePropertyQName:
 - 783 ○ A resource property QName does not identify a proper number of resource
 - 784 ○ properties.
 - 785 • SetResourcePropertyRequestFailed
 - 786 ○ One or more components of the SetResourceProperties request failed.
- 787 OtherFaults: tbd
- 788 Note: All faults generated must be compliant with the WS-BaseFaults [WS-BaseFaults]
- 789 specification.
- 790 The fault message MUST also indicate whether the effects of processing previous components
- 791 were restored or not. Note: There is no isolation policy implied, for either modifications to the
- 792 resource properties document resulting from the processing of the request or the modifications
- 793 implemented by the restore. See Section 7 for more discussion.
- 794

795 5.3.1 Example SOAP Encoding of the SetResourceProperties

796 Message Exchange

797 Consider the following resource properties document defining resource properties for a WS-

798 Resource defined by the GenericDiskDrive portType:

```
799 <GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >
800   <tns:NumberOfBlocks>22</tns:NumberOfBlocks>
801   <tns:BlockSize>1024</tns:BlockSize>
802   <tns:Manufacturer>DrivesRUs</tns:Manufacturer>
803 </GenericDiskDriveProperties>
```

804 The following is a non-normative example of a SetResourceProperties request message using

805 SOAP 1.2 [SOAP 1.2]:

```
806 <s12:Envelope
807   xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
808   xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
809   xmlns:wsrp=
810     "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-
811 ResourceProperties-1.2-draft-01.xsd"
812   xmlns:ex="http://example.com/exampleNS">
813   <s12:Header>
814     <wsa:Action>
815       http://docs.oasis-open.org/wsrp/2004/06/WS-
816 ResourceProperties/SetResourceProperties
817     </wsa:Action>
818     <wsa:To s12:mustUnderstand="1">
819       http://www.provider.org/ProviderEndpoint
```

```

820     </wsa:To>
821     <ex:ResourceDisambiguator>
822         uuid:84dec55-7d3f-65ad-ac44-675d9fce5d22
823     </ex:ResourceDisambiguator>
824 </s12:Header>
825 <s12:Body>
826     <wsrpw:SetResourceProperties
827         xmlns:tns="http://example.com/diskdrive">
828         <wsrp:Update>
829             <tns:NumberOfBlocks>143</tns:NumberOfBlocks>
830         </wsrp:Update>
831
832         <wsrp>Delete resourceProperty="tns:Manufacturer" />
833
834         <wsrp:Insert>
835             <tns:someElement>42</tns:someElement>
836         </wsrp:Insert>
837
838     </wsrpw:SetResourceProperties>
839 </s12:Body>
840 </s12:Envelope>

```

841 The following is an example SetResourcePropertiesResponse message using SOAP 1.2 [SOAP
842 1.2]:

```

843 <s12:Envelope
844     xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
845     xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
846     xmlns:wsrp=
847 "http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-
848 1.2-draft-01.xsd"
849     xmlns:resp="http://www.other.org/otherNS">
850 <s12:Header>
851     <wsa:Action>
852         http://docs.oasis-open.org/wsrf/2004/06/WS-
853 ResourceProperties/SetResourcePropertiesResponse
854     </wsa:Action>
855     <wsa:To s12:mustUnderstand="1">
856         http://www.requestor.org/someEndpoint
857     </wsa:To>
858     <resp:SomeResourceRef>
859         uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
860     </resp:SomeResourceRef>
861 </s12:Header>
862 <s12:Body>
863     <wsrp:SetResourcePropertiesResponse>
864     </wsrp:SetResourcePropertiesResponse>
865 </s12:Body>
866 </s12:Envelope>

```

867 The new contents of the resource properties document after successful processing of the request
868 message MUST be:

```
869 <GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >  
870   <tns:NumberOfBlocks>143</tns:NumberOfBlocks>  
871   <tns:BlockSize>1024</tns:BlockSize>  
872   <tns:someElement>42</tns:someElement>  
873 </GenericDiskDriveProperties>
```

874 5.4 QueryResourceProperties

875 A Web service that implements a portType that includes the resource properties document type
876 declaration (/wsdl:portType/@ResourceProperties) is a component of a WS-Resource, and MAY
877 support the message exchange defined in this section that allows a requestor to query the
878 resource properties document of a WS-Resource using a query expression such as XPath
879 [XPath].

880 The format of this request message MUST be:

```
881 <wsrp:QueryResourceProperties>  
882   <wsrp:QueryExpression Dialect="URI">  
883     xsd:any  
884   </wsrp:QueryExpression>  
885 </wsrp:QueryResourceProperties>
```

886 The QueryResourceProperties request message MUST follow the implied resource pattern, as
887 defined in Section 2.

888 The components of the QueryResourceProperties request message are further described as
889 follows:

890 /wsrp:QueryResourceProperties/wsrp:QueryExpression

891 The context of the expression is to be evaluated against the resource properties
892 document of the WS-Resource identified by the request. The results of evaluating the
893 QueryExpression are returned in the response to this request message.

894 /wsrp:QueryResourceProperties/wsrp:QueryExpression/@Dialect

895 This attribute contains a URI specifying the type of expression contained by the element.
896 If the implementation does not recognize the URI identified by @Dialect, it MUST fault.
897 There are two well known dialects identified by this specification, corresponding to two
898 versions of the XPath language.

899 <http://www.w3.org/TR/1999/REC-xpath-19991116>

900 This URI identifies the XPath 1.0 language. The contents of the
901 QueryExpression MUST be a string containing a valid XPath 1.0
902 expression.

903 <http://www.w3.org/TR/2003/WD-xpath20-20031112>

904 This URI identifies the Xpath 2.0 (working draft) language. The contents
905 of the QueryExpression MUST be a string containing a valid XPath 2.0
906 expression. Note: an additional URI will be added to represent the W3C
907 Recommendation form of the XPath 2.0 language.

908 /wsrp:QueryResourceProperties/QueryExpression/{any}

909 The QueryExpression MUST contain an expression in an expression language specified
910 by the dialect attribute. Note: this element may contain mixed content.

911 The response of the QueryResourceProperties request message MUST be a message of the
912 following form:

```
913 <wsrp:QueryResourcePropertiesResponse>  
914 {any}  
915 </wsrp:QueryResourcePropertiesResponse>
```

916 The contents of the QueryResourcePropertiesResponse message are further described as
917 follows:

918 /wsrp:QueryResourcePropertiesResponse/{any}

919 The response of the QueryResourceProperties request is variable, depending on the
920 nature of the QueryExpression component of the QueryResourceProperties request. The
921 response MUST contain an XML serialization of the results of evaluating the
922 QueryExpression against the resource properties document. Note: this element has
923 mixedContent, to allow for the case where the QueryExpression evaluates to a simple
924 type (such as a Boolean, a string or an integer) as well as the case where a node-set of
925 elements is returned.

926 If the WS-Resource does not respond to the QueryResourceProperties request message with the
927 QueryResourcePropertiesResponse message, then it MUST send one of the following fault
928 messages:

- 929 • ResourceUnknownFault
 - 930 ○ The resource identified in the message (which follows the implied resource pattern) is
931 not known to the Web service.
- 932 • UnknownQueryExpressionDialect
 - 933 ○ The given QueryExpression has a dialect that is unknown to the Web service.
- 934 • InvalidQueryExpression
 - 935 ○ The given Query Expression is not valid within the QueryExpression language
936 identified by the dialect attribute.
- 937 • QueryEvaluationError
 - 938 ○ The Query Expression failed during evaluation.

939 OtherFaults: tbd

940 Note: All faults generated must be compliant with the WS-BaseFaults [WS-BaseFaults]
941 specification.

942 **5.4.1 Example SOAP Encoding of the QueryResourceProperties** 943 **Message Exchange**

944 Consider the following resource properties document defining resource properties for a WS-
945 Resource defined by the GenericDiskDrive portType:

```
946 <GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >  
947 <tns:NumberOfBlocks>22</tns:NumberOfBlocks>  
948 <tns:BlockSize>1024</tns:BlockSize>  
949 <tns:Manufacturer>DrivesRUs</tns:Manufacturer>  
950 </GenericDiskDriveProperties>
```

951 The following is a non-normative example of a QueryResourceProperties request message using
952 SOAP 1.2 [SOAP 1.2]:

```
953 <s12:Envelope
954     xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
955     xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
956     xmlns:wsrp=
957 "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-
958 1.2-draft-01.xsd"
959     xmlns:ex="http://example.com/exampleNS">
960   <s12:Header>
961     <wsa:Action>
962       http://docs.oasis-open.org/wsrp/2004/06/WS-
963 ResourceProperties/QueryResourceProperties
964     </wsa:Action>
965     <wsa:To s12:mustUnderstand="1">
966       http://www.provider.org/ProviderEndpoint
967     </wsa:To>
968     <ex:ResourceDisambiguator>
969       uuid:84dec55-7d3f-65ad-ac44-675d9fce5d22
970     </ex:ResourceDisambiguator>
971   </s12:Header>
972   <s12:Body>
973     <wsrp:QueryResourceProperties>
974       <wsrp:QueryExpression
975         Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116" >
976         boolean(/*/NumberOfBlocks > 20 and */BlockSize=1024)
977       </wsrp:QueryExpression>
978     </wsrp:QueryResourceProperties>
979   </s12:Body>
980 </s12:Envelope>
```

981 The following is an example QueryResourcePropertiesResponse message using SOAP 1.2
982 [SOAP 1.2], containing the results of evaluating that XPath expression against the root element of
983 the resource's resource properties document:

```
984 <s12:Envelope
985     xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
986     xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
987     xmlns:wsrp=
988 "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-
989 1.2-draft-01.xsd"
990     xmlns:resp="http://www.other.org/otherNS">
991   <s12:Header>
992     <wsa:Action>
993       http://docs.oasis-open.org/wsrp/2004/06/WS-
994 ResourceProperties/QueryResourcePropertiesResponse
995     </wsa:Action>
996     <wsa:To s12:mustUnderstand="1">
997       http://www.requestor.org/someEndpoint
```

```
998     </wsa:To>
999     <resp:SomeResourceRef>
1000         uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
1001     </resp:SomeResourceRef>
1002 </s12:Header>
1003 <s12:Body>
1004     <wsrp:QueryResourcePropertiesResponse>
1005         true
1006     </wsrp:QueryResourcePropertiesResponse>
1007 </s12:Body>
1008 </s12:Envelope>
```

6 Subscription

1009

1010 The WS-Notification [WS-Notification] family of specifications describes the patterns, concepts,
1011 standard message exchanges, and protocols of a topic-based, publish-subscribe messaging
1012 pattern in Web services. In the notification model, a service creates messages that are delivered
1013 to other services that had previously registered interest in the situation associated with that
1014 message.

1015 With WS-ResourceProperties, it is a common pattern for Web service requestors to request
1016 notification of changes (inserts, updates and deletions) made to the values of one or more
1017 resource property elements of a given WS-Resource. It is the Web service component of the WS-
1018 Resource that is responsible for executing or observing the messages. This suggests the need
1019 for encapsulation of the stateful resource to ensure all changes made to the stateful resource are
1020 *observed* by the WS-Resource implementation. To the extent that encapsulation is not provided,
1021 and updates to the stateful resource are made outside of the knowledge of the associated WS-
1022 Resource implementation, the WS-Resource may not be able to provide notifications reflecting
1023 those updates.

1024 If a WS-Resource supports the resource property value-change notification pattern, and if it uses
1025 WS-Notification to implement this feature, then it **MUST** implement the message exchanges for
1026 the NotificationProducer role, as specified in [WS-BaseNotification]. The WS-Resource **MAY**
1027 accept subscriptions to only a subset of the resource properties defined for a WS-Resource. If an
1028 implementation does not use WS-Notification, then it **MAY** ignore the requirements outlined in this
1029 section.

1030 One notification message artifact is created for each change to each resource property observed
1031 by the WS-Resource implementation. For example, a SetResourceProperties request message
1032 may contain five SetRequestComponents. Each of these components would result in the creation
1033 of a separate message artifact.

1034 WS-ResourceProperties defines the Notification Topic and TopicSpace elements [WS-Topics]
1035 that **MUST** be used to express the organization of the WS-Resource property element value
1036 change notifications. By understanding the relationship between Topics and resource properties,
1037 and examining the set of Topics supported by the NotificationProducer Web service, the service
1038 requestor can determine which of the resource properties are able to participate in the value-
1039 change notification pattern. The Topic and TopicSpace elements associated with resource
1040 property value-change notification are described as follows:

- 1041 1. The WS-Resource's resource properties document **MAY** be defined using resource
1042 properties declared in multiple XML namespaces. For each of these XML namespaces,
1043 an associated TopicSpace element **MUST** be defined. The TopicSpace element defines
1044 a topic space intended to contain topics related to value changes of resource properties
1045 declared in that XML namespace.
 - 1046 ○ The value of the TopicSpace element's targetNamespace attribute **MUST** be the
1047 same as the URI of the namespace in which the resource property element is
1048 defined. The name attribute of the TopicSpace element **SHOULD** have the value
1049 "ResourcePropertiesTopicSpace".
- 1050 2. For each resource property participating in the value-change notification pattern, a Topic
1051 element **MUST** be defined as a child of the TopicSpace element defined in 1.
 - 1052 ○ Notification messages reflecting changes to the resource property are associated
1053 with this Topic.

- 1054 ○ The value of the Topic element's name attribute MUST be the same as the
1055 NCName of the resource property element.
- 1056 ○ The value of the Topic element's messageTypes attribute MUST include
1057 wsrp:ResourcePropertyValueChangeNotification (defined later in this section). In
1058 addition, it MAY include QNames of other message elements.
- 1059 ○ A designer MAY introduce additional child sub-topic elements to the topic
1060 element that represent application-specific needs.
- 1061 3. The WS-Resource acting as the NotificationProducer MUST include Topics defined in 2,
1062 as part of the value of its "Topics" resource property element. One such Topic MUST be
1063 included for each resource property element offered as a target for a value-change
1064 subscription.
- 1065 4. When a WS-Resource observes a resource property value change, it SHOULD create a
1066 notification message that expresses the situation, and associate the notification message
1067 with the Topic associated with that resource property. Note: there are many
1068 circumstances in which a change to a resource property might not result in the generation
1069 of a notification message. For example, a resource property value may change
1070 frequently, making generation of notification messages too expensive for the service. In
1071 this situation, a WS-Resource may choose to never generate notification message
1072 artifacts to record value change, or it may choose to generate notification message
1073 artifacts for a subset of the value change situations.

1074 The wsrp:ResourcePropertyValueChangeNotification element MUST appear as a component of
1075 the notification message associated with resource property value change topics. This element is
1076 defined as follows:

```
1077       <wsrp:ResourcePropertyValueChangeNotification>  
1078        <wsrp:OldValue> xsd:any </wsrp:OldValue>?  
1079        <wsrp:NewValue> xsd:any </wsrp:NewValue>  
1080       </wsrp:ResourcePropertyValueChangeNotification>
```

1081 This element may appear as the root element of the notification message, or it may appear as a
1082 descendent of the root, accommodating patterns where the notification message itself is
1083 contained in an enveloping mechanism. The form of the
1084 ResourcePropertyValueChangeNotification is further constrained as follows:

1085 /wsrp:ResourcePropertyValueChangeNotification

1086 One ResourcePropertyValueChangeNotification element is created for each resource
1087 property value change situation detected and acted upon by the WS-Resource. This
1088 component records the value change of the affected resource property.

1089 /wsrp:ResourcePropertyValueChangeNotification/OldValue

1090 This element, if it appears, MUST contain the value of the affected WS-Resource
1091 property immediately prior to when the value change was applied. If the resource
1092 property did not have a value prior to the value change (for example, this notification
1093 represents an insertion of a new resource property element) then this element is empty
1094 and will contain the attribute xsi:nil with value "true". If this value does not appear in the
1095 message, then the WS-Resource was unable or unwilling to record the value prior to the
1096 value change.

1097 /wsrp:ResourcePropertyValueChangeNotification/NewValue

1098 This element MUST contain the value of the affected WS-Resource property after the
1099 value change condition was detected. If the WS-Resource property does not have a
1100 value after the value change (for example, this notification represents a deletion of the
1101 resource property element) then this element is empty and will contain the attribute xsi:nil
1102 with value "true".

1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129

7 ACID Properties of Operations on WS-Resources

The ability to associate a transactional recovery policy to the execution of a Web service message exchange is described in the Web Services Atomic Transaction specification [WS-AtomicTransaction]. In the presence of a transactional unit of work, a Web service capable of participating in the transactional protocol must abide by the rules of two-phase-commit transaction management. However, in the absence of a transaction management policy, the Web service is under no obligation to recover the state of the WS-Resource in the event of a failure during message processing.

This specification is not prescriptive with respect to policy that governs concurrent read or write access to a WS-Resource. The definition of specific policy governing concurrent updates, whether or not separate message executions targeting the same WS-Resource may be interleaved, and whether partially completed WS-Resource updates within a given message execution may be observed by other concurrent requests is beyond the scope of this definition. The scope and extent of the isolation of changes made to the WS-Resource is an implementation dependent responsibility of the WS-Resource itself. The WS-Resource must also take on the responsibility for the scope and extent to which notifications of changes to the WS-Resource are isolated and made observable. If WS-Resource update isolation is needed, we suggest the use of a transaction [WS-AtomicTransaction] to provide a context within which isolation of WS-Resource updates can be provided. In the absence of a transactional unit of work, the level of WS-Resource update atomicity, recovery, isolation, and durability provided is implementation-dependent.

The ability to declare and attach isolation-level policy to the definition of a Web service message exchange, whether or not a transactional unit of work is present, represents a general requirement not met by the current Web service architecture. In the future, isolation-level policy declarations may be introduced as a formal part of the WS-Resource definition. Refer to [State Paper] for a general discussion of these requirements.

1130 8 Security Considerations

1131 This specification defines the resource properties document and also the set of message
1132 exchanges that MUST be supported by a WS-Resource. In this context, there are two categories
1133 of security aspects that need to be considered: (a) securing the message exchanges and (b)
1134 securing the resource properties.

1135 8.1 Securing the message exchanges

1136 When messages are exchanged between a requestor and a WS-Resource in order to access or
1137 act on one or more resource properties, it is RECOMMENDED that the communication between
1138 services be secured using the mechanisms described in WS-Security. In order to properly secure
1139 messages, the message body and all relevant headers need to be included in the digital
1140 signature so as to prove the integrity of the message. In addition the ReferenceProperties from an
1141 EndpointReference, used as part of any message exchange, may be encrypted to ensure their
1142 privacy. In the event that a requestor communicates frequently with a Web service to access
1143 resource properties, either directly through a query or accomplished through notification of state
1144 change, it is RECOMMENDED that a security context be established using the mechanisms
1145 described in WS-Trust [WS-Trust] and WS-SecureConversation [WS-SecureConversation],
1146 allowing for potentially more efficient means of authentication.

1147 It is common for communication between requestors and the WS-Resource to exchange multiple
1148 messages. As a result, the usage profile may be susceptible to key attacks. For this reason it is
1149 RECOMMENDED that the keys used to secure the channel be changed frequently. This "re-
1150 keying" can be effected a number of ways. The following list outlines four common techniques:

- 1151 • Attaching a nonce to each message and using it in a derived key function with the shared
1152 secret
- 1153 • Using a derived key sequence and switch "generations"
- 1154 • Closing and re-establishing a security context
- 1155 • Exchanging new secrets between the parties

1156 It should be noted that the mechanisms listed above are independent of the security context
1157 token (SCT). That is, the keys used to secure the channel during message exchanges may be
1158 independent of the key used to prove the right to access WS-ResourceProperties.

1159 The security context MAY be re-established using the mechanisms described in WS-Trust and
1160 WS-SecureConversation. Similarly, secrets can be exchanged using the mechanisms described
1161 in WS-Trust. Note, however, that the current shared secret SHOULD NOT be used to encrypt the
1162 new shared secret. Derived keys, the preferred solution from this list, can be specified using the
1163 mechanisms described in WS-SecureConversation.

1164 The following list summarizes common classes of attacks that apply to this protocol and identifies
1165 the mechanism to prevent/mitigate the attacks:

- 1166 • **Message alteration** – Alteration is prevented by including signatures of the message
1167 information using WS-Security.
- 1168 • **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-
1169 Security.
- 1170 • **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by
1171 comparing secured policies – see WS-Policy [WS-Policy] and WS-SecurityPolicy [WS-
1172 SecurityPolicy]).

- 1173 • **Authentication** – Authentication is established using the mechanisms described in WS-
1174 Security and WS-Trust. Each message is authenticated using the mechanisms described in
1175 WS-Security.
- 1176 • **Accountability** – Accountability is a function of the type of and string of the key and
1177 algorithms being used. In many cases, a strong symmetric key provides sufficient
1178 accountability. However, in some environments, strong PKI signatures are required.
- 1179 • **Availability** – Many services are subject to a variety of availability attacks. Replay is a
1180 common attack and it is RECOMMENDED that this be addressed as described in the Replay
1181 bullet item below. Other attacks, such as network-level denial of service attacks, are harder to
1182 avoid and are outside the scope of this specification. That said, care should be taken to
1183 ensure that minimal processing be performed prior to any authenticating sequences.
- 1184 • **Replay** – Messages may be replayed for a variety of reasons. To detect and eliminate this
1185 attack, mechanisms should be used to identify replayed messages such as the
1186 timestamp/nonce outlined in WS-Security and the sequences outlined in WS-
1187 ReliableMessaging [WS-ReliableMessaging].

1188 8.2 Securing Resource Properties

1189 Since WS-ResourceProperties defines a mechanism to expose properties of a WS-Resource,
1190 security policies should be established that ensure that only authorized requestors can access
1191 the value of a resource property. In order to secure access to the resource properties, the
1192 message exchanges that provide the access should be appropriately controlled. Authorization
1193 policies should be put in place so that the implications of providing the state information (through
1194 GetResourceProperty, GetMultipleResourceProperties, or QueryResourceProperties messages
1195 or through notification of value change and modification of the resource properties), are taken into
1196 account. These policies should also take into account the semantic difference between
1197 components of the SetResourceProperties message – i.e. that an Update component updates a
1198 *value* of a resource property, whereas Insert and Delete components modify whether the WS-
1199 Resource actually *contains* the resource property values.

1200 The authorization policies may also reflect the sensitivity of the resource property(ies) that are
1201 accessible from a WS-Resource. Policies can be set at the coarse granularity of the message
1202 exchange (e.g., Get(Multiple)ResourceProperty(ies) vs SetResourceProperty), but finer-grained
1203 control at the level of individual resource properties may be desired in some scenarios (e.g. user
1204 Bob can access value of “Manufacturer” but not “NumberOfBlocks”).

1205 Given that a requestor will be able to access a resource property value by subscribing to state
1206 changes, care should be taken to set up security policies so that a consistent policy is in effect
1207 irrespective of whether the resource property value is accessed through direct message
1208 exchanges (e.g., GetResourceProperty) or indirectly through subscription for state changes (i.e.,
1209 subscription to “ResourceChangePropertyValueNotification” topic). It should also be noted that a
1210 requestor will be able to query the value of a property through the QueryResourceProperty
1211 operation, or by using a domain-specific operation corresponding to a resource property (e.g.,
1212 getNumberOfBlocks) if one exists. Therefore, the authorization policy on QueryResourceProperty
1213 operation (and the getXXX operation, if one is declared on the Web service for resource property
1214 named XXX) should be set so that a requestor who is not authorized to get a value of a resource
1215 property through a GetResourceProperty request is not able to deduce the value indirectly
1216 through the QueryResourceProperty request (or the getXXX operation on the Web service).

1217 Even if the requestor is authorized to access the requested resource properties, it is
1218 RECOMMENDED that the resource properties that are exchanged between a requestor and a

- 1219 Web service are secured to ensure integrity and/or confidentiality of the resource property values.
- 1220 This will prevent unauthorized alteration of and/or access to the property values while in transit.
- 1221 This would mean that the specific resource property elements are signed and/or encrypted within
- 1222 the message by leveraging WS-Security as discussed in the previous section.

1223 9 References

1224 9.1 Normative

- 1225 [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 1226
- 1227
- 1228 [SOAP 1.2] <http://www.w3.org/TR/soap12-part1/>
- 1229 [URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.
- 1230
- 1231
- 1232 [State Paper] [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/wsrf/download.php/6795/ws-modelingresources.pdf)
- 1233 [open.org/apps/org/workgroup/wsrf/download.php/6795/ws-](http://www.oasis-open.org/apps/org/workgroup/wsrf/download.php/6795/ws-modelingresources.pdf)
- 1234 [modelingresources.pdf](http://www.oasis-open.org/apps/org/workgroup/wsrf/download.php/6795/ws-modelingresources.pdf)
- 1235 [WS-Addressing] <http://www.ibm.com/developerworks/webservices/library/ws-add/>
- 1236 [WS-BaseNotification] [http://docs.oasis-open.org/wsn/2004/06/wsn-WS-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)
- 1237 [BaseNotification-1.2-draft-03.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf)
- 1238 [WS-Policy] <http://www.ibm.com/developerworks/library/ws-policy>
- 1239 [WS-ResourceLifetime] [http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-](http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-03.pdf)
- 1240 [ResourceLifetime-1.2-draft-03.pdf](http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-03.pdf)
- 1241 [WS-Topics] [http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)
- 1242 [draft-01.pdf](http://docs.oasis-open.org/wsn/2004/06/wsn-WS-Topics-1.2-draft-01.pdf)
- 1243 [XML-Infoset] <http://www.w3.org/TR/xml-infoset/>
- 1244 [XPath] <http://www.w3.org/TR/xpath>

1245 9.2 Non-Normative

- 1246 [OGSI 1.0] Open Grid Services Infrastructure (OGSI) V1.0
- 1247 [http://forge.gridforum.org/projects/ggf-editor/document/draft-ogsi-](http://forge.gridforum.org/projects/ggf-editor/document/draft-ogsi-service-1/en/1)
- 1248 [service-1/en/1](http://forge.gridforum.org/projects/ggf-editor/document/draft-ogsi-service-1/en/1)
- 1249 [WS-AtomicTransaction] [http://www.ibm.com/developerworks/webservices/library/ws-](http://www.ibm.com/developerworks/webservices/library/ws-atomtran/)
- 1250 [atomtran/](http://www.ibm.com/developerworks/webservices/library/ws-atomtran/)
- 1251 [WS-Notification] [http://www.oasis-](http://www.oasis-open.org/apps/org/workgroup/wsn/download.php/6661/WSNpubsub-1-0.pdf)
- 1252 [open.org/apps/org/workgroup/wsn/download.php/6661/WSNpub](http://www.oasis-open.org/apps/org/workgroup/wsn/download.php/6661/WSNpubsub-1-0.pdf)
- 1253 [sub-1-0.pdf](http://www.oasis-open.org/apps/org/workgroup/wsn/download.php/6661/WSNpubsub-1-0.pdf)
- 1254 [WS-ReliableMessaging] <http://www.ibm.com/developerworks/webservices/library/ws-rm/>
- 1255 [WS-SecureConversation] <http://www.ibm.com/developerworks/library/ws-secon/>
- 1256 [WS-Security] [http://www.oasis-](http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf)
- 1257 [open.org/committees/download.php/5531/oasis-200401-wss-](http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf)
- 1258 [soap-message-security-1.0.pdf](http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf)
- 1259 [WS-SecurityPolicy] <http://www.ibm.com/developerworks/library/ws-secpol/>
- 1260 [WS-Trust] <http://www.ibm.com/developerworks/library/ws-trust/>
- 1261 [WSDL 2.0] <http://www.w3.org/TR/wsdl12/>

1262 **Appendix A. Acknowledgments**

1263 Special thanks to the Global Grid Forum's Open Grid Services Infrastructure working group,
1264 which defined the OGSi v1.0 [OGSI] specification which was a large inspiration for the ideas
1265 expressed in this specification.

1266 The following individuals were members of the committee during the development of this
1267 specification:

1268

1269 Akhil Arora (Sun Microsystems), Tim Banks (IBM), Jeff Bohren (OpenNetwork), Conor Cahill
1270 (AOL), Fred Carter (AmberPoint), Martin Chapman (Oracle), Glen Daniels (Sonic Software),
1271 Thomas Freund (IBM), Stephen Graham (IBM), Anish Karmarkar (Oracle), Hideharu Kato
1272 (Hitachi), David Levine (IBM), Paul Lipton (Computer Associates), Mark Little (Arjuna
1273 Technologies Limited), Lily Liu (WebMethods, Inc.), Tom Maguire (IBM), Susan Malaika (IBM),
1274 David Martin (IBM), Samuel Meder (ArgonneNational Laboratory), Jeff Mischkinsky (Oracle),
1275 Bryan Murray (Hewlett-Packard), Dave Orchard (BEA Systems, Inc.), Savas Parastatidis
1276 (Individual), Greg Pavlik (Oracle), Mark Peel (Novell), Alain Regnier (Ricoh Company, Ltd.), Ian
1277 Robinson (IBM), Junaid Saiyed (Sun Microsystems), Igor Sedukhin (Computer Associates),
1278 Hitoshi Sekine (Ricoh Company, Ltd.), Frank Siebenlist (ArgonneNational Laboratory), David
1279 Snelling (Fujitsu), Latha Srinivasan (Hewlett-Packard), John Tollefsrud (Sun Microsystems), Jem
1280 Treadwell (Hewlett-Packard), Steve Tuecke (ArgonneNational Laboratory), William Vambenepe
1281 (Hewlett-Packard), Katy Warr (IBM), Alan Weissberger (NEC Corporation), and Pete Wenzel
1282 (SeeBeyond Technology Corporation)

1283

1284 In addition, the following people made contributions to this specification:

1285 Nick Butler (IBM), Karl Czajkowski (Globus / USC/ISI), Andrew Eisenberg (IBM), Donald F
1286 Ferguson (IBM), Ian Foster (Globus / Argonne), Jeffrey Frey (IBM), Diane Jordan (IBM), Frank
1287 Leymann (IBM), Andreas Meier (IBM), Nataraj Nagaratnam (IBM), Martin Nally (IBM), John
1288 Rofrano (IBM), Ellen Stokes (IBM), Tony Storey (IBM), Jay Unger (IBM), Sanjiva Weerawarana
1289 (IBM).

1290 Appendix B. XML Schema

1291 The XML types and elements used in this specification are defined in the following XML Schema:

```
1292 <?xml version="1.0" encoding="UTF-8"?>
1293 <!--
1294
1295 OASIS takes no position regarding the validity or scope of any
1296 intellectual property or other rights that might be claimed to pertain
1297 to the implementation or use of the technology described in this
1298 document or the extent to which any license under such rights might or
1299 might not be available; neither does it represent that it has made any
1300 effort to identify any such rights. Information on OASIS's procedures
1301 with respect to rights in OASIS specifications can be found at the
1302 OASIS website. Copies of claims of rights made available for
1303 publication and any assurances of licenses to be made available, or the
1304 result of an attempt made to obtain a general license or permission for
1305 the use of such proprietary rights by implementors or users of this
1306 specification, can be obtained from the OASIS Executive Director.
1307
1308 OASIS invites any interested party to bring to its attention any
1309 copyrights, patents or patent applications, or other proprietary rights
1310 which may cover technology that may be required to implement this
1311 specification. Please address the information to the OASIS Executive
1312 Director.
1313
1314 Copyright (C) OASIS Open (2004). All Rights Reserved.
1315
1316 This document and translations of it may be copied and furnished to
1317 others, and derivative works that comment on or otherwise explain it or
1318 assist in its implementation may be prepared, copied, published and
1319 distributed, in whole or in part, without restriction of any kind,
1320 provided that the above copyright notice and this paragraph are
1321 included on all such copies and derivative works. However, this
1322 document itself may not be modified in any way, such as by removing the
1323 copyright notice or references to OASIS, except as needed for the
1324 purpose of developing OASIS specifications, in which case the
1325 procedures for copyrights defined in the OASIS Intellectual Property
1326 Rights document must be followed, or as required to translate it into
1327 languages other than English.
1328
1329 The limited permissions granted above are perpetual and will not be
1330 revoked by OASIS or its successors or assigns.
1331
```

1332 This document and the information contained herein is provided on an
1333 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1334 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
1335 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1336 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
1337
1338 -->
1339 <xsd:schema
1340 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1341 xmlns:wsrp=
1342 "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-
1343 1.2-draft-01.xsd"
1344 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1345 elementFormDefault="qualified" attributeFormDefault="unqualified"
1346 targetNamespace=
1347 "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-
1348 1.2-draft-01.xsd"
1349 >
1350
1351 <!-- ===== Global Attribute Declaration for WSDL 1.1 portType===== -->
1352 <xsd:attribute name="ResourceProperties" type="xsd:QName" />
1353
1354 <!-- = Notification Message for ResourceProperties value change === -->
1355 <xsd:complexType name="ResourcePropertyValueChangeNotificationType">
1356 <xsd:sequence>
1357 <xsd:element name="OldValue" nillable="true"
1358 minOccurs="0" maxOccurs="1" >
1359 <xsd:complexType>
1360 <xsd:sequence>
1361 <xsd:any minOccurs="1" maxOccurs="1" />
1362 </xsd:sequence>
1363 </xsd:complexType>
1364 </xsd:element>
1365 <xsd:element name="NewValue" nillable="true"
1366 minOccurs="1" maxOccurs="1" >
1367 <xsd:complexType>
1368 <xsd:sequence>
1369 <xsd:any minOccurs="1" maxOccurs="1" />
1370 </xsd:sequence>
1371 </xsd:complexType>
1372 </xsd:element>
1373 </xsd:sequence>
1374 </xsd:complexType>
1375
1376 <xsd:element name="ResourcePropertyValueChangeNotification"
1377 type="wsrp:ResourcePropertyValueChangeNotificationType" />
1378
1379 <xsd:complexType name="QueryExpressionType" mixed="true">

```
1380     <xsd:sequence>
1381         <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
1382     </xsd:sequence>
1383     <xsd:attribute name="Dialect" type="xsd:anyURI" />
1384 </xsd:complexType>
1385
1386     <xsd:element name="QueryExpression" type="wsrp:QueryExpressionType"
1387 />
1388
1389 </xsd:schema>
```

1390

Appendix C. WSDL 1.1

1391
1392

The following illustrates the WSDL 1.1 for the Web service methods described in this specification:

1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432

```
<?xml version="1.0" encoding="utf-8"?>
<!--
OASIS takes no position regarding the validity or scope of any
intellectual property or other rights that might be claimed to pertain
to the implementation or use of the technology described in this
document or the extent to which any license under such rights might or
might not be available; neither does it represent that it has made any
effort to identify any such rights. Information on OASIS's procedures
with respect to rights in OASIS specifications can be found at the
OASIS website. Copies of claims of rights made available for
publication and any assurances of licenses to be made available, or the
result of an attempt made to obtain a general license or permission for
the use of such proprietary rights by implementors or users of this
specification, can be obtained from the OASIS Executive Director.
OASIS invites any interested party to bring to its attention any
copyrights, patents or patent applications, or other proprietary rights
which may cover technology that may be required to implement this
specification. Please address the information to the OASIS Executive
Director.
Copyright (C) OASIS Open (2004). All Rights Reserved.
This document and translations of it may be copied and furnished to
others, and derivative works that comment on or otherwise explain it or
assist in its implementation may be prepared, copied, published and
distributed, in whole or in part, without restriction of any kind,
provided that the above copyright notice and this paragraph are
included on all such copies and derivative works. However, this
document itself may not be modified in any way, such as by removing the
copyright notice or references to OASIS, except as needed for the
purpose of developing OASIS specifications, in which case the
procedures for copyrights defined in the OASIS Intellectual Property
Rights document must be followed, or as required to translate it into
languages other than English.
The limited permissions granted above are perpetual and will not be
revoked by OASIS or its successors or assigns.
```

1433 This document and the information contained herein is provided on an
1434 "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED,
1435 INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
1436 INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
1437 WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
1438
1439 -->
1440
1441 <wsdl:definitions name="WS-ResourceProperties"
1442 xmlns="http://schemas.xmlsoap.org/wsdl/"
1443 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
1444 xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
1445 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1446 xmlns:wsbf=
1447 "http://docs.oasis-open.org/wsr/2004/06/wsr/WS-BaseFaults-1.2-
1448 draft-01.xsd"
1449 xmlns:wsrp=
1450 "http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-
1451 1.2-draft-01.xsd"
1452 xmlns:wsrpw=
1453 "http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-
1454 1.2-draft-01.wsdl"
1455 targetNamespace=
1456 "http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-
1457 1.2-draft-01.wsdl"
1458 >
1459
1460 <!-- ===== Types Definitions ===== -->
1461 <wsdl:types>
1462 <xsd:schema
1463 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1464 targetNamespace=
1465 "http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-
1466 1.2-draft-01.xsd"
1467 elementFormDefault="qualified"
1468 attributeFormDefault="unqualified">
1469
1470 <xsd:include schemaLocation=
1471 "http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-
1472 1.2-draft-01.xsd"
1473 />
1474
1475 <xsd:import
1476 namespace=
1477 "http://schemas.xmlsoap.org/ws/2003/03/addressing"
1478 schemaLocation=
1479 "http://schemas.xmlsoap.org/ws/2003/03/addressing"
1480 />

```

1481
1482     <xsd:import
1483         namespace=
1484         "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-BaseFaults-1.2-
1485 draft-01.xsd"
1486         schemaLocation=
1487         "http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-BaseFaults-1.2-
1488 draft-01.xsd"
1489     />
1490
1491 <!-- ===== Message Types for GetResourceProperty ===== -->
1492
1493     <xsd:element name="GetResourceProperty"
1494         type="xsd:QName" />
1495
1496     <xsd:element name="GetResourcePropertyResponse" >
1497         <xsd:complexType>
1498             <xsd:sequence>
1499                 <xsd:any minOccurs="0" maxOccurs="unbounded" />
1500             </xsd:sequence>
1501         </xsd:complexType>
1502     </xsd:element>
1503
1504     <xsd:complexType name="ResourceUnknownFaultType">
1505         <xsd:complexContent>
1506             <xsd:extension base="wsbf:BaseFaultType"/>
1507         </xsd:complexContent>
1508     </xsd:complexType>
1509     <xsd:element name="ResourceUnknownFault"
1510         type="wsrp:ResourceUnknownFaultType"/>
1511
1512     <xsd:complexType name="InvalidResourcePropertyQNameFaultType">
1513         <xsd:complexContent>
1514             <xsd:extension base="wsbf:BaseFaultType"/>
1515         </xsd:complexContent>
1516     </xsd:complexType>
1517     <xsd:element name="InvalidResourcePropertyQNameFault"
1518         type="wsrp:InvalidResourcePropertyQNameFaultType"/>
1519
1520
1521 <!-- ===== Message Types for GetMultipleResourceProperties ===== -->
1522     <xsd:element name="GetMultipleResourceProperties">
1523         <xsd:complexType>
1524             <xsd:sequence>
1525                 <xsd:element name="ResourceProperty" type="xsd:QName"
1526                     minOccurs="1" maxOccurs="unbounded" />
1527             </xsd:sequence>
1528         </xsd:complexType>

```

```

1529     </xsd:element>
1530
1531     <xsd:element name="GetMultipleResourcePropertiesResponse">
1532         <xsd:complexType>
1533             <xsd:sequence>
1534                 <xsd:any minOccurs="0" maxOccurs="unbounded" />
1535             </xsd:sequence>
1536         </xsd:complexType>
1537     </xsd:element>
1538
1539 <!-- ===== Message Types for SetResourceProperties ===== -->
1540
1541     <xsd:complexType name="InsertType">
1542         <xsd:sequence>
1543             <xsd:any processContents="lax"
1544                 minOccurs="1" maxOccurs="unbounded" />
1545         </xsd:sequence>
1546     </xsd:complexType>
1547     <xsd:element name="Insert"
1548         type="wsrp:InsertType" />
1549
1550     <xsd:complexType name="UpdateType">
1551         <xsd:sequence>
1552             <xsd:any processContents="lax"
1553                 minOccurs="1" maxOccurs="unbounded" />
1554         </xsd:sequence>
1555     </xsd:complexType>
1556     <xsd:element name="Update"
1557         type="wsrp:UpdateType" />
1558
1559     <xsd:complexType name="DeleteType">
1560         <xsd:attribute name="ResourceProperty"
1561             type="xsd:QName" use="required" />
1562     </xsd:complexType>
1563     <xsd:element name="Delete"
1564         type="wsrp>DeleteType" />
1565
1566     <xsd:element name="SetResourceProperties">
1567         <xsd:complexType>
1568             <xsd:choice minOccurs="0" maxOccurs="unbounded">
1569                 <xsd:element ref="wsrp:Insert" />
1570                 <xsd:element ref="wsrp:Update" />
1571                 <xsd:element ref="wsrp>Delete" />
1572             </xsd:choice>
1573         </xsd:complexType>
1574     </xsd:element>
1575
1576     <xsd:element name="SetResourcePropertiesResponse" >

```

```

1577         <xsd:complexType />
1578     </xsd:element>
1579
1580     <xsd:complexType
1581 name="InvalidSetResourcePropertiesRequestContentFaultType">
1582         <xsd:complexContent>
1583             <xsd:extension base="wsbf:BaseFaultType" />
1584         </xsd:complexContent>
1585     </xsd:complexType>
1586 <xsd:element
1587 name="InvalidSetResourcePropertiesRequestContentFault"
1588 type="wsrp:InvalidSetResourcePropertiesRequestContentFaultType" />
1589
1590     <xsd:complexType name="UnableToModifyResourcePropertyFaultType">
1591         <xsd:complexContent>
1592             <xsd:extension base="wsbf:BaseFaultType" />
1593         </xsd:complexContent>
1594     </xsd:complexType>
1595 <xsd:element name="UnableToModifyResourcePropertyFault"
1596 type="wsrp:UnableToModifyResourcePropertyFaultType" />
1597
1598     <xsd:complexType
1599 name="SetResourcePropertyRequestFailedFaultType">
1600         <xsd:complexContent>
1601             <xsd:extension base="wsbf:BaseFaultType" />
1602         </xsd:complexContent>
1603     </xsd:complexType>
1604 <xsd:element name="SetResourcePropertyRequestFailedFault"
1605 type="wsrp:SetResourcePropertyRequestFailedFaultType" />
1606
1607 <!-- ===== Message Types for QueryResourceProperties ===== -->
1608
1609     <xsd:element name="QueryResourceProperties" >
1610         <xsd:complexType>
1611             <xsd:sequence>
1612                 <xsd:element ref="wsrp:QueryExpression"
1613                     minOccurs="1" maxOccurs="1" />
1614             </xsd:sequence>
1615         </xsd:complexType>
1616     </xsd:element>
1617
1618     <xsd:element name="QueryResourcePropertiesResponse" >
1619         <xsd:complexType>

```

```

1625         <xsd:complexContent mixed="true">
1626             <xsd:restriction base="xsd:anyType">
1627                 <xsd:sequence>
1628                     <xsd:any processContents="lax"
1629                         minOccurs="1" maxOccurs="unbounded" />
1630                 </xsd:sequence>
1631             </xsd:restriction>
1632         </xsd:complexContent>
1633     </xsd:complexType>
1634 </xsd:element>
1635
1636 <xsd:complexType name="UnknownQueryExpressionDialectFaultType">
1637     <xsd:complexContent>
1638         <xsd:extension base="wsbf:BaseFaultType" />
1639     </xsd:complexContent>
1640 </xsd:complexType>
1641 <xsd:element name="UnknownQueryExpressionDialectFault"
1642     type="wsrp:UnknownQueryExpressionDialectFaultType" />
1643
1644 <xsd:complexType name="InvalidQueryExpressionFaultType">
1645     <xsd:complexContent>
1646         <xsd:extension base="wsbf:BaseFaultType" />
1647     </xsd:complexContent>
1648 </xsd:complexType>
1649 <xsd:element name="InvalidQueryExpressionFault"
1650     type="wsrp:InvalidQueryExpressionFaultType" />
1651
1652 <xsd:complexType name="QueryEvaluationErrorFaultType">
1653     <xsd:complexContent>
1654         <xsd:extension base="wsbf:BaseFaultType" />
1655     </xsd:complexContent>
1656 </xsd:complexType>
1657 <xsd:element name="QueryEvaluationErrorFault"
1658     type="wsrp:QueryEvaluationErrorFaultType" />
1659
1660
1661 </xsd:schema>
1662 </wsdl:types>
1663
1664 <!-- ===== GetResourceProperty =====>
1665 GetResourceProperty(QName)
1666 returns: any
1667 -->
1668 <wsdl:message name="GetResourcePropertyRequest">
1669     <wsdl:part name="GetResourcePropertyRequest"
1670         element="wsrp:GetResourceProperty" />
1671 </wsdl:message>
1672

```

```

1673 <wsdl:message name="GetResourcePropertyResponse">
1674   <wsdl:part name="GetResourcePropertyResponse"
1675     element="wsrp:GetResourcePropertyResponse" />
1676 </wsdl:message>
1677
1678 <wsdl:message name="ResourceUnknownFault">
1679   <part name="ResourceUnknownFault"
1680     element="wsrp:ResourceUnknownFault" />
1681 </wsdl:message>
1682
1683 <wsdl:message name="InvalidResourcePropertyQNameFault">
1684   <part name="InvalidResourcePropertyQNameFault"
1685     element="wsrp:InvalidResourcePropertyQNameFault" />
1686 </wsdl:message>
1687
1688 <!-- =====GetMultipleResourceProperties =====
1689   GetMultipleResourceProperties(list of QName)
1690   returns: sequence of any
1691 -->
1692 <wsdl:message name="GetMultipleResourcePropertiesRequest">
1693   <wsdl:part name="GetMultipleResourcePropertiesRequest"
1694     element="wsrp:GetMultipleResourceProperties" />
1695 </wsdl:message>
1696
1697 <wsdl:message name="GetMultipleResourcePropertiesResponse">
1698   <wsdl:part name="GetMultipleResourcePropertiesResponse"
1699     element="wsrp:GetMultipleResourcePropertiesResponse" />
1700 </wsdl:message>
1701
1702 <!-- ===== SetResourceProperties =====
1703   SetResourceProperties(
1704     { insert (any)* |
1705       update (any)* |
1706       delete@QName } +
1707   )
1708   returns: empty
1709 -->
1710 <wsdl:message name="SetResourcePropertiesRequest">
1711   <wsdl:part name="SetResourcePropertiesRequest"
1712     element="wsrp:SetResourceProperties" />
1713 </wsdl:message>
1714
1715 <wsdl:message name="SetResourcePropertiesResponse">
1716   <wsdl:part name="SetResourcePropertiesResponse"
1717     element="wsrp:SetResourcePropertiesResponse" />
1718 </wsdl:message>
1719
1720 <wsdl:message name="InvalidSetResourcePropertiesRequestContentFault">

```

```

1721     <part name="InvalidSetResourcePropertiesRequestContentFault"
1722
1723 element="wsrp:InvalidSetResourcePropertiesRequestContentFault" />
1724 </wsdl:message>
1725
1726 <wsdl:message name="UnableToModifyResourcePropertyFault">
1727     <part name="UnableToModifyResourcePropertyFault"
1728         element="wsrp:UnableToModifyResourcePropertyFault" />
1729 </wsdl:message>
1730
1731 <wsdl:message name="SetResourcePropertyRequestFailedFault">
1732     <part name="SetResourcePropertyRequestFailedFault"
1733         element="wsrp:SetResourcePropertyRequestFailedFault" />
1734 </wsdl:message>
1735
1736 <!-- ===== QueryResourceProperties =====
1737 QueryResourceProperties(QueryExpression)
1738 returns: any
1739 -->
1740 <wsdl:message name="QueryResourcePropertiesRequest">
1741     <wsdl:part name="QueryResourcePropertiesRequest"
1742         element="wsrp:QueryResourceProperties" />
1743 </wsdl:message>
1744
1745 <wsdl:message name="QueryResourcePropertiesResponse">
1746     <wsdl:part name="QueryResourcePropertiesResponse"
1747         element="wsrp:QueryResourcePropertiesResponse" />
1748 </wsdl:message>
1749
1750 <wsdl:message name="UnknownQueryExpressionDialectFault">
1751     <part name="UnknownQueryExpressionDialectFault"
1752         element="wsrp:UnknownQueryExpressionDialectFault" />
1753 </wsdl:message>
1754
1755 <wsdl:message name="InvalidQueryExpressionFault">
1756     <part name="InvalidQueryExpressionFault"
1757         element="wsrp:InvalidQueryExpressionFault" />
1758 </wsdl:message>
1759
1760 <wsdl:message name="QueryEvaluationErrorFault">
1761     <part name="QueryEvaluationErrorFault"
1762         element="wsrp:QueryEvaluationErrorFault" />
1763 </wsdl:message>
1764
1765 <!-- ===== PortType Definitions ===== -->
1766 <wsdl:portType name="GetResourceProperty">
1767     <wsdl:operation name="GetResourceProperty">
1768         <wsdl:input name="GetResourcePropertyRequest"

```

```

1769         message="wsrpw:GetResourcePropertyRequest" />
1770     <wsdl:output name="GetResourcePropertyResponse"
1771         message="wsrpw:GetResourcePropertyResponse" />
1772     <wsdl:fault name="ResourceUnknownFault"
1773         message="wsrpw:ResourceUnknownFault" />
1774     <wsdl:fault name="InvalidResourcePropertyQNameFault"
1775         message="wsrpw:InvalidResourcePropertyQNameFault" />
1776 </wsdl:operation>
1777 </wsdl:portType>
1778
1779 <wsdl:portType name="GetMultipleResourceProperties">
1780     <wsdl:operation name="GetMultipleResourceProperties">
1781         <wsdl:input name="GetMultipleResourcePropertiesRequest"
1782             message="wsrpw:GetMultipleResourcePropertiesRequest"
1783         />
1784         <wsdl:output name="GetMultipleResourcePropertiesResponse"
1785             message="wsrpw:GetMultipleResourcePropertiesResponse" />
1786         <wsdl:fault name="ResourceUnknownFault"
1787             message="wsrpw:ResourceUnknownFault" />
1788         <wsdl:fault name="InvalidResourcePropertyQNameFault"
1789             message="wsrpw:InvalidResourcePropertyQNameFault" />
1790     </wsdl:operation>
1791 </wsdl:portType>
1792
1793 <wsdl:portType name="SetResourceProperties">
1794     <wsdl:operation name="SetResourceProperties">
1795         <wsdl:input name="SetResourcePropertiesRequest"
1796             message="wsrpw:SetResourcePropertiesRequest" />
1797         <wsdl:output name="SetResourcePropertiesResponse"
1798             message="wsrpw:SetResourcePropertiesResponse" />
1799         <wsdl:fault name="ResourceUnknownFault"
1800             message="wsrpw:ResourceUnknownFault" />
1801         <wsdl:fault
1802             name="InvalidSetResourcePropertiesRequestContentFault"
1803             message="wsrpw:InvalidSetResourcePropertiesRequestContentFault" />
1804         <wsdl:fault name="UnableToModifyResourcePropertyFault"
1805             message="wsrpw:UnableToModifyResourcePropertyFault"
1806         />
1807         <wsdl:fault name="InvalidResourcePropertyQNameFault"
1808             message="wsrpw:InvalidResourcePropertyQNameFault" />
1809         <wsdl:fault name="SetResourcePropertyRequestFailedFault"
1810             message="wsrpw:SetResourcePropertyRequestFailedFault" />
1811     </wsdl:operation>
1812 </wsdl:portType>
1813
1814 </wsdl:operation>
1815 </wsdl:portType>
1816

```

```
1817 <wsdl:portType name="QueryResourceProperties">
1818   <wsdl:operation name="QueryResourceProperties">
1819     <wsdl:input name="QueryResourcePropertiesRequest"
1820       message="wsrpw:QueryResourcePropertiesRequest" />
1821     <wsdl:output name="QueryResourcePropertiesResponse"
1822       message="wsrpw:QueryResourcePropertiesResponse" />
1823     <wsdl:fault name="ResourceUnknownFault"
1824       message="wsrpw:ResourceUnknownFault" />
1825     <wsdl:fault name="InvalidResourcePropertyQNameFault"
1826       message="wsrpw:InvalidResourcePropertyQNameFault" />
1827     <wsdl:fault name="UnknownQueryExpressionDialectFault"
1828       message="wsrpw:UnknownQueryExpressionDialectFault"
1829   />
1830     <wsdl:fault name="InvalidQueryExpressionFault"
1831       message="wsrpw:InvalidQueryExpressionFault" />
1832     <wsdl:fault name="QueryEvaluationErrorFault"
1833       message="wsrpw:QueryEvaluationErrorFault" />
1834   </wsdl:operation>
1835 </wsdl:portType>
1836 </wsdl:definitions>
```

Appendix D. Revision History

Rev	Date	By Whom	What
wd-01	2004-05-18	Steve Graham	Initial version created from submission by contributing companies. Minor modifications made to reflect OASIS formatting and the following issues: WSRF2, WSRF3, WSRF14, WSRF33.
wd-02	2004-05-31	Steve Graham, Jem Treadwell	Mods to draft 01, including hyphenation, clarification of acknowledgements section
wd-03	2004-06-04	Steve Graham	Reformat rogue Veranda text with Arial.
wd-04	2004-06-07	Steve Graham	Base faults comment on faults (align with ResourceLifetime), update date URIs to 2004/06, update URLs in references to point to .pdfs, update Acknowledgements

1841 **Appendix E. Notices**

1842 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
1843 that might be claimed to pertain to the implementation or use of the technology described in this
1844 document or the extent to which any license under such rights might or might not be available;
1845 neither does it represent that it has made any effort to identify any such rights. Information on
1846 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
1847 website. Copies of claims of rights made available for publication and any assurances of licenses
1848 to be made available, or the result of an attempt made to obtain a general license or permission
1849 for the use of such proprietary rights by implementors or users of this specification, can be
1850 obtained from the OASIS Executive Director.

1851

1852 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
1853 applications, or other proprietary rights which may cover technology that may be required to
1854 implement this specification. Please address the information to the OASIS Executive Director.

1855

1856 Copyright (C) OASIS Open (2004). All Rights Reserved.

1857

1858 This document and translations of it may be copied and furnished to others, and derivative works
1859 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
1860 published and distributed, in whole or in part, without restriction of any kind, provided that the
1861 above copyright notice and this paragraph are included on all such copies and derivative works.
1862 However, this document itself may not be modified in any way, such as by removing the copyright
1863 notice or references to OASIS, except as needed for the purpose of developing OASIS
1864 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
1865 Property Rights document must be followed, or as required to translate it into languages other
1866 than English.

1867

1868 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
1869 successors or assigns.

1870

1871 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1872 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
1873 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
1874 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
1875 PARTICULAR PURPOSE.