

Web Services Resource Properties (WS-ResourceProperties)

Version 1.1

03/05/2003

Authors

Steve Graham (IBM) (Editor)

Karl Czajkowski (Globus / USC/ISI)

Donald F Ferguson (IBM)

Ian Foster (Globus / Argonne)

Jeffrey Frey (IBM)

Frank Leymann (IBM)

Tom Maguire (IBM)

Nataraj Nagaratnam (IBM)

Martin Nally (IBM)

Tony Storey (IBM)

Igor Sedukhin (Computer Associates International)

David Snelling (Fujitsu Laboratories of Europe)

Steve Tuecke (Globus / Argonne)

William Vambenepe (Hewlett-Packard)

Sanjiva Weerawarana (IBM)

Copyright Notice

© Copyright Computer Associates International Inc., Fujitsu Limited, Hewlett-Packard Development Company, International Business Machines Corporation and The University of Chicago 2003, 2004. All Rights Reserved.

Permission to copy and display this "Web Services Resource Properties" Specification ("this Specification"), in any medium without fee or royalty is hereby granted, provided that you include the following on ALL copies of this Specification, or portions thereof, that you make:

1. A link or URL to this Specification at this location.
2. This Copyright Notice as shown in this Specification.

THIS SPECIFICATION IS PROVIDED "AS IS," AND COMPUTER ASSOCIATES INTERNATIONAL, FUJITSU LIMITED, HEWLETT-PACKARD DEVELOPMENT COMPANY, IBM AND THE UNIVERSITY OF CHICAGO (COLLECTIVELY, THE "COMPANIES") MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE; THAT THE CONTENTS OF THIS SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL

NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE COMPANIES WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS SPECIFICATION.

The companies each agree to grant you a royalty-free license, under commercially reasonable terms and conditions, to their respective patents that they deem necessary to implement this Specification.

The names and trademarks of the Companies may NOT be used in any manner, including advertising or publicity pertaining to this Specification or its contents, without specific, written prior permission. Title to copyright in this Specification will at all times remain with the Companies.

No other rights are granted by implication, estoppel or otherwise.

PORTIONS OF THIS MATERIAL WERE PREPARED AS AN ACCOUNT OF WORK SPONSORED BY IBM CORPORATION AT UNIVERSITY OF CHICAGO'S ARGONNE NATIONAL LABORATORY. NEITHER THE AUTHORS, NOR THE UNITED STATES GOVERNMENT OR ANY AGENCY THEREOF, NOR THE UNIVERSITY OF CHICAGO, NOR IBM, NOR ANY OF THEIR EMPLOYEES OR OFFICERS, NOR ANY OTHER COPYRIGHT HOLDERS OR CONTRIBUTORS, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS. REFERENCE HEREIN TO ANY SPECIFIC COMMERCIAL PRODUCT, PROCESS, OR SERVICE BY TRADE NAME, TRADEMARK, MANUFACTURER, OR OTHERWISE, DOES NOT NECESSARILY CONSTITUTE OR IMPLY ITS ENDORSEMENT, RECOMMENDATION, OR FAVORING BY IBM, THE UNITED STATES GOVERNMENT OR ANY AGENCY THEREOF OR ANY OTHER COPYRIGHT HOLDERS OR CONTRIBUTORS. THE VIEW AND OPINIONS OF AUTHORS EXPRESSED HEREIN DO NOT NECESSARILY STATE OR REFLECT THOSE OF IBM, THE UNITED STATES GOVERNMENT OR ANY AGENCY THEREOF, OR THE ENTITY BY WHICH AN AUTHOR MAY BE EMPLOYED.

This manuscript has been created in part by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

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 WS-ResourceProperties specification is an initial draft release and is provided for review and evaluation only. The Companies hope to solicit your contributions and suggestions in the near future. The Companies make no warranties or representations regarding the specification in any manner whatsoever.

Table of Contents

1	INTRODUCTION	5
1.1	GOALS AND REQUIREMENTS	5
1.1.1	<i>Requirements</i>	5
1.1.2	<i>Non-Goals</i>	6
1.2	NOTATIONAL CONVENTIONS.....	6
1.3	NAMESPACES	6
2	TERMINOLOGY AND CONCEPTS	7
3	EXAMPLE	8
4	DECLARING RESOURCE PROPERTIES	9
4.1	RESOURCE PROPERTIES DOCUMENT.....	9
4.2	RESOURCE PROPERTIES DOCUMENT TYPE	10
4.3	DECLARING THE RESOURCE PROPERTIES DOCUMENT TYPE IN WSDL	10
4.4	RESOURCE PROPERTIES AND INTERFACE AGGREGATION	11
5	OPERATIONS ON RESOURCE PROPERTIES	13
5.1	GETRESOURCEPROPERTY	13
5.1.1	<i>Example SOAP Encoding of the GetResourceProperty Message Exchange</i>	14
5.2	GETMULTIPLERESOURCEPROPERTIES.....	15
5.2.1	<i>Example SOAP Encoding of the GetMultipleResourceProperties Message Exchange</i>	17
5.3	SETRESOURCEPROPERTIES.....	18
5.3.1	<i>Example SOAP Encoding of the SetResourceProperties Message Exchange</i>	21
5.4	QUERYRESOURCEPROPERTIES	22
5.4.1	<i>Example SOAP Encoding of the QueryResourceProperties Message Exchange</i>	24
6	SUBSCRIPTION	25
7	ACID PROPERTIES OF OPERATIONS ON WS-RESOURCES	27
8	SECURITY CONSIDERATIONS	28
8.1	SECURING THE MESSAGE EXCHANGES	28
8.2	SECURING RESOURCE PROPERTIES.....	29
9	ACKNOWLEDGEMENTS	30
10	REFERENCES	30
11	APPENDIX I – XML SCHEMA	31
12	APPENDIX II – WSDL 1.1	33

1 Introduction

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 include a component that identifies a stateful resource to be used in the execution of the message exchange. We refer to the composition of a stateful resource and a Web service under the implied resource pattern as a WS-Resource.

This specification standardizes the means by which the definition of the properties of a WS-Resource may be declared as part of the Web service interface. The declaration of the WS-Resource's properties represents a projection of or a *view* on the WS-Resource's state. The 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 the Web service interface.

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

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

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

1.1 Goals and Requirements

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

1.1.1 Requirements

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

This specification MUST:

- Define the term "resource property" and its relationship to Web services and WS-Resources.
- Define the means by which a designer decorates a Web service description with the names and types of properties associated with a WS-Resource.
- Define the means by which a requestor can:
 - Retrieve the values of one or more properties of a WS-Resource
 - Update the values of one or more properties of a WS-Resource

- Query across the values of a one or more properties of a Ws-Resource
- Subscribe for notification [WS-Notification] when the value of a WS-Resource property changes.

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

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

1.1.2 Non-Goals

The following topics are outside the scope of this specification:

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

1.2 Notational Conventions

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

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

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

1.3 Namespaces

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 (this spec)	http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties
wsnt	http://www.ibm.com/xmlns/stdwip/web-services/WS-BaseNotification

2 Terminology and Concepts

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

WS-Resource:

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

Implied Resource Pattern:

- The way WS-Addressing must be used to designate the stateful resource component of the WS-Resource to be used in the execution of message exchanges.
- An EndpointReference that follows the implied resource pattern may include a ReferenceProperties child element that identifies the stateful resource component of the WS-Resource to be used in the execution of all message exchanges performed using this EndpointReference.
- A message that follows the implied resource pattern **MUST** be sent to a Web service referred to by an EndpointReference that follows the implied resource pattern, and **MUST** conform to the WS-Addressing requirements on that message including adding the ReferenceProperties information, if present, from that EndpointReference to the message.
- A Web service that follows the implied resource pattern **MAY** use the ReferenceProperties information from a message that follows the implied resource pattern in order to identify the stateful resource to be used in the execution requested by that message.

WS-Resource Qualified Endpoint Reference

- An Endpoint Reference used to refer to a WS-Resource composed of a Web service and a stateful resource.
- A stateful resource identifier **MAY** be contained within the ReferenceProperties element of the Endpoint Reference.

- The address of the Web service associated with the WS-Resource MUST be contained in the Address element of the Endpoint Reference.

Resource Property:

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

Resource Properties Document:

- The XML document representing a logical composition of resource property elements. The resource properties document defines a particular view or projection of the state data implemented by the WS-Resource.
- The *type* (e.g. the XML Schema definition of the root element) of a resource properties document is associated with the WSDL portType defining the Web service interface. This association is the basis of the WS-Resource definition. All instances of a particular WS-Resource type MUST implement a logical resource properties document of the type declared in the WSDL portType.

Resource Property Element:

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

Resource Property Value:

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

3 Example

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

```
<wsdl:definitions ... xmlns:tns="http://example.com/diskDrive" ...>
...
<wsdl:types>
  <xsd:schema targetNamespace="http://example.com/diskDrive" ... >

    <!-- Resource property element declarations -->
    <xsd:element name="NumberOfBlocks" type="xsd:integer"/>
    <xsd:element name="BlockSize" type="xsd:integer" />
    <xsd:element name="Manufacturer" type="xsd:string" />

    <!-- Resource properties document declaration -->
    <xsd:element name="GenericDiskDriveProperties">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:NumberOfBlocks"/>
          <xsd:element ref="tns:BlockSize" />
          <xsd:element ref="tns:Manufacturer" />
          <xsd:any minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
```



```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
...
</xsd:schema>
</wsdl:types>
...
<!-- Association of resource properties document to a portType -->
<wsdl:portType name="GenericDiskDrive"
    wsrp:ResourceProperties="tns:GenericDiskDriveProperties" >

    <operation name="start" .../>
    <operation name="stop" .../>
...
</wsdl:portType>
...
</wsdl:definitions>

```

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

```

...
<wsrp:GetMultipleResourceProperties
    xmlns:tns="http://example.com/diskdrive" ...>
    <wsrp:ResourceProperty>tns:NumberOfBlocks</wsrp:ResourceProperty>
    <wsrp:ResourceProperty>tns:BlockSize</wsrp:ResourceProperty>
</wsrp:GetMultipleResourceProperties>
...

```

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

```

...
<wsrp:GetMultipleResourcePropertiesResponse
    xmlns:ns1="http://example.com/diskdrive" ...>
    <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
    <ns1:BlockSize>1024</ns1:BlockSize>
</wsrp:GetMultipleResourcePropertiesResponse>
...

```

4 Declaring Resource Properties

4.1 Resource Properties Document

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

This specification does not dictate the means by which a service implements a resource properties document. Some service implementations may choose to realize its implementation of the resource properties document as an actual XML instance document, stored in memory, in the file system, in a database or in some XML Repository. Other service implementations may *dynamically* construct the resource property elements and their values, from data held in programming language objects

(such as a J2EE EJB Entity Bean) or by executing a command on a private communications channel to a physical resource. Yet another implementation possibility is a mapping layer to a standard management interface (such as CIM or SNMP).

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

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

4.2 Resource Properties Document Type

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

1. The resource properties document MUST be a global element declaration (GED) in some XML namespace. This GED defines the type of the root element of a resource properties document and hence the type of the resource properties document itself.
2. The resource properties document MUST be uniquely identified by a QName.
3. The complexType defining the resource properties document MUST define element children only; it MUST not define attributes. The child elements MUST be aggregated using `xsd:sequence` or `xsd:all`. The order of appearance of the resource properties within the resource properties document does not matter to WS-ResourceProperties.
4. The complexType defining the resource properties document MUST define a sequence of one or more child elements, called *resource property elements*.
 - a. Child elements MUST be defined using XML schema element reference (`@ref`).
 - b. This specification defines no additional restriction on the use of `@minOccurs` or `@maxOccurs` or other information elements associated with the XML Schema element definition.
5. The complexType defining the resource properties document MAY allow open element content (`xsd:any`).

4.3 Declaring the Resource Properties Document Type in WSDL

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

```

<wsdl:definitions ...>
  <wsdl:portType ...
    wsrp:ResourceProperties="xsd:QName"? ... >
  ...
</wsdl:portType>

```

This definition is further constrained as follows:

/wsdl:portType/@wsrp:ResourceProperties

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

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

4.4 Resource Properties and Interface Aggregation

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

Within WSDL 1.1, there is no formally defined interface extension mechanism¹. In WSDL 1.1 we expect service designers to *copy-and-paste* operations from the various constituent interfaces into a single, flat, most derived service interface. In addition, we expect the service interface designer to compose a resource property document for the most derived Web service interface that consists of all of the resource property element declarations from each of the constituent interfaces used in the composition.

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

```

<wsdl:definitions ...
  xmlns:gen="http://example.com/diskDrive"
  xmlns:ven="http://vendor.com/diskDrive"
  ...>
...
<wsdl:types>
  <xsd:schema targetNamespace="http://vendor.com/diskDrive" ... >

    <!-- Resource property element declarations -->
    <xsd:element name="VendorExtension" type="xsd:string" />

```

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

```

<!-- Resource properties document declaration -->
<xsd:element name="VendorDiskDriveProperties">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="gen:NumberOfBlocks"/>
      <xsd:element ref="gen:BlockSize" />
      <xsd:element ref="gen:Manufacturer" />
      <xsd:element ref="ven:VendorExtension" />
      <xsd:any minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
...
</xsd:schema>
</wsdl:types>
...
<!-- Association of resource properties document to a portType -->
<wsdl:portType name="VendorDiskDrive"
  wsrp:ResourceProperties="ven:VendorDiskDriveProperties" >
  <operation name="...
...
  <!-- copy/paste operations from genericDiskDrive -->
  <operation name="start" .../>
  <operation name="stop" .../>
...
  <!-- define Vendor-specific operations -->
  <operation name="reset" .../>
...
</wsdl:portType>
...
</wsdl:definitions>

```

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

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

1. Define the new portType.

In this example the new portType is named "VendorDiskDrive", this portType extends "GenericDiskDrive".
2. Copy all of the operation child elements from the portType being extended, and paste them as child elements of the new portType; the order of the operations SHOULD be preserved.

In this example, the "start" and "stop" operations are copied from the GenericDiskDrive portType and pasted as child elements of the VendorDiskDrive portType.
3. Define additional, vendor-specific operations as child elements of the new portType.

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

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

In this example, the element is named "VendorDiskDriveProperties" and defined in the "http://vendor.com/diskDrive" namespace.
5. Copy all of the child elements (@ref and xsd:any) from the resource properties document of the portType being extended, and paste them as child elements of the new resource properties document; the order of the elements SHOULD be preserved. This step MUST be repeated for each portType that is being extended by this new portType. Any duplicate child elements MUST be removed.

In this example, the elements that reference (@ref) "gen:NumberOfBlocks", "gen:Blocksize", and "gen:Manufacturer" and the "xsd:any" are copied from the GenericDiskDriveProperties declaration and pasted to the VendorDiskDriveProperties declaration.
6. Define any additional resource property elements that are specific to the newly defined resource properties document type.

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

5 Operations on Resource Properties

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

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

5.1 GetResourceProperty

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

The format of this request message MUST be:

```
<wsrp:GetResourceProperty>
  QName
</wsrp:GetResourceProperty>
```

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

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

/wsrp:GetResourceProperty/QName

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

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

```
<wsrp:GetResourcePropertyResponse>
  {any}*
</wsrp:GetResourcePropertyResponse>
```

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

/wsrp:GetResourcePropertyResponse/{any}

The resource property value, as an XML element, that corresponds to the QName in the GetResourceProperty request.

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

- ResourceUnknownFault
 - The resource identified in the message (which follows the implied resource pattern) is not known to the Web service.
- InvalidResourcePropertyQName
 - The QName in the request message did not correspond to a resource property element of the WS-Resource referred to in the request message.
- OtherFaults: tbd

5.1.1 Example SOAP Encoding of the GetResourceProperty Message Exchange

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

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

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:ex="http://example.com/exampleNS">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
      ResourceProperties/GetResourceProperty
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.provider.org/ProviderEndpoint
```

```

    </wsa:To>
    <ex:ResourceId>
      uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
    </ex:ResourceId>
  </s12:Header>
  <s12:Body>
    <wsrp:GetResourceProperty
      xmlns:tns="http://example.com/diskDrive">
      tns:NumberOfBlocks
    </wsrp: GetResourceProperty>
  </s12:Body>
</s12:Envelope>

```

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

```

<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:resp="http://www.other.org/otherNS">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties/GetResourcePropertyResponse
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.requestor.org/someEndpoint
    </wsa:To>
    <resp:SomeResourceId>
      uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
    </resp:SomeResourceId>
  </s12:Header>
  <s12:Body>
    <wsrp:GetResourcePropertyResponse
      xmlns:ns1="http://example.com/diskDrive">
      <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
    </wsrp:GetResourcePropertyResponse>
  </s12:Body>
</s12:Envelope>

```

5.2 GetMultipleResourceProperties

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

The format of this request message MUST be:

```

<wsrp:GetMultipleResourceProperties>
  <wsrp:ResourceProperty>QName <wsrp:ResourceProperty>+
</wsrp:GetMultipleResourceProperties>

```

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

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

/wsrp:GetMultipleResourceProperties/wsrp:ResourceProperty+

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

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

```
<wsrp:GetMultipleResourcePropertiesResponse>
  {any}*
</wsrp:GetMultipleResourcePropertiesResponse>
```

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

/wsrp:GetMultipleResourcePropertiesResponse/{any}

A collection of resource property values, as XML elements that correspond to the QNames given in the GetMultipleResourceProperties request message. This collection is formed in the following fashion. For each QName in the request message, the resource must add to the collection all child elements of the root of the resource properties document whose name corresponds to that QName. If the XML schema definition of the resource properties document root element does not permit the root element to contain a child element with that QName the processing of the GetMultipleResourceProperties request message MUST terminate with a fault message.

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

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

- ResourceUnknownFault
 - The resource identified in the message (which follows the implied resource pattern) is not known to the Web service.
- InvalidResourcePropertyQName
 - One or more of the QNames in the request message did not correspond to a resource property element of the WS-Resource referred to in the request message.
- OtherFaults: tbd

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

efficient retrieval of multiple resource property values using a single message exchange.

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

5.2.1 Example SOAP Encoding of the `GetMultipleResourceProperties` Message Exchange

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

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

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:ex="http://example.com/exampleNS">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
      ResourceProperties/GetMultipleResourceProperties
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.provider.org/ProviderEndpoint
    </wsa:To>
    <ex:ResourceId>
      uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
    </ex:ResourceId>
  </s12:Header>
  <s12:Body>
    <wsrp:GetMultipleResourceProperties
      xmlns:tns="http://example.com/diskdrive">
      <wsrp:ResourceProperty>tns:NumberOfBlocks<wsrp:ResourceProperty>
      <wsrp:ResourceProperty>tns:BlockSize<wsrp:ResourceProperty>
    </wsrp:GetMultipleResourceProperties>
  </s12:Body>
</s12:Envelope>
```

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:resp="http://www.other.org/otherNS">
  <s12:Header>
    <wsa:Action>
```

```

    http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties/GetMultipleResourcePropertiesResponse
  </wsa:Action>
  <wsa:To s12:mustUnderstand="1">
    http://www.requestor.org/someEndpoint
  </wsa:To>
  <resp:SomeResourceId>
    uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
  </resp:SomeResourceId>
</s12:Header>
<s12:Body>
  <wsrp:GetMultipleResourcePropertiesResponse
    xmlns:ns1="http://example.com/diskdrive" ...>
    <ns1:NumberOfBlocks>22</ns1:NumberOfBlocks>
    <ns1:BlockSize>1024</ns1:BlockSize>
  </wsrp:GetMultipleResourcePropertiesResponse>
</s12:Body>
</s12:Envelope>

```

5.3 SetResourceProperties

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

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

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

The format of this request message MUST be:

```

<wsrp:SetResourceProperties>
  {
  <wsrp:Insert >
    {any}*
  </wsrp:Insert> |

  <wsrp:Update >
    {any}*
  </wsrp:Update> |

  <wsrp>Delete ResourceProperty="QName" />
  }+
</wsrp:SetResourceProperties>

```

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

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

/wsrp:SetResourceProperties

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

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

/wsrp:SetResourceProperties/wsrp:Insert

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

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

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

/wsrp:SetResourceProperties/wsrp:Update

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

given name. In such circumstances, the resource MUST fault the processing of the component.

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

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

/wsrp: SetResourceProperties/wsrp:Delete

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

/wsrp: SetResourceProperties/wsrp:Delete/@ResourceProperty

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

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

```
<wsrp:SetResourcePropertiesResponse>
</wsrp:SetResourcePropertiesResponse>
```

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

- ResourceUnknownFault
 - The resource identified in the message (which follows the implied resource pattern) is not known to the Web service.
- InvalidSetResourcePropertiesRequestContent:
 - The *contents* of the SetResourceProperties request component causes the resource properties document to no longer validate.
- UnableToModifyResourceProperty:
 - A resource property identified by one of the SetResourceProperties request components is read-only.
- InvalidResourcePropertyQName:

- A resource property QName does not identify a proper number of resource properties.
- SetResourcePropertyRequestFailed
 - One or more components of the SetResourceProperties request failed.
- OtherFaults: tbd

The fault message MUST also indicate whether the effects of processing previous components were restored or not. Note: There is no isolation policy implied, for either modifications to the resource properties document resulting from the processing of the request or the modifications implemented by the restore.

5.3.1 Example SOAP Encoding of the SetResourceProperties Message Exchange

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

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

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:ex="http://example.com/exampleNS">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties/SetResourceProperties
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.provider.org/ProviderEndpoint
    </wsa:To>
    <ex:ResourceId>
      uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
    </ex:ResourceId>
  </s12:Header>
  <s12:Body>
    <wsrp:SetResourceProperties
      xmlns:tns="http://example.com/diskdrive">
      <wsrp:Update resourceProperty="tns:NumberOfBlocks">
        <tns:NumberOfBlocks>143</tns:NumberOfBlocks>
      </wsrp:Update>

      <wsrp>Delete resourceProperty="tns:Manufacturer" />

      <wsrp:Insert>
        <tns:someElement>42</tns:someElement>
      </wsrp:Insert>
```

```

    </wsrp:SetResourceProperties>
  </s12:Body>
</s12:Envelope>

```

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

```

<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:resp="http://www.other.org/otherNS">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties/SetResourcePropertiesResponse
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.requestor.org/someEndpoint
    </wsa:To>
    <resp:SomeResourceId>
      uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
    </resp:SomeResourceId>
  </s12:Header>
  <s12:Body>
    <wsrp:SetResourcePropertiesResponse>
    </wsrp:SetResourcePropertiesResponse>
  </s12:Body>
</s12:Envelope>

```

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

```

<GenericDiskDriveProperties xmlns:tns="http://example.com/diskDrive" >
  <tns:NumberOfBlocks>143</tns:NumberOfBlocks>
  <tns:BlockSize>1024</tns:BlockSize>
  <tns:someElement>42</tns:someElement>
</GenericDiskDriveProperties>

```

5.4 QueryResourceProperties

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

The format of this request message MUST be:

```

<wsrp:QueryResourceProperties>
  <wsrp:QueryExpression dialect="URI">
    xsd:any
  </wsrp:QueryExpression>
</wsrp:QueryResourceProperties>

```

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

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

/wsrp:QueryResourceProperties/wsrp:QueryExpression

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

/wsrp:QueryResourceProperties/wsrp:QueryExpression/@dialect

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

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

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

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

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

/wsrp:QueryResourceProperties/QueryExpression/{any}

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

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

```
<wsrp:QueryResourcePropertiesResponse>
  {any}
</wsrp:QueryResourcePropertiesResponse>
```

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

/wsrp:QueryResourcePropertiesResponse/{any}

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

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

- ResourceUnknownFault
 - The resource identified in the message (which follows the implied resource pattern) is not known to the Web service.

- UnknownQueryExpressionDialect
 - The given QueryExpression has a dialect that is unknown to the Web service.
- InvalidQueryExpression
 - The given Query Expression is not a valid within the QueryExpression language identified by the dialect attribute. .
- QueryEvaluationError
 - The Query Expression failed during evaluation.
- OtherFaults: tbd

5.4.1 Example SOAP Encoding of the QueryResourceProperties Message Exchange

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

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

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
  xmlns:wsrp="
    http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:ex="http://example.com/exampleNS">
  <s12:Header>
    <wsa:Action>
      http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties/QueryResourceProperties
    </wsa:Action>
    <wsa:To s12:mustUnderstand="1">
      http://www.provider.org/ProviderEndpoint
    </wsa:To>
    <ex:ResourceId>
      uuid:84decd55-7d3f-65ad-ac44-675d9fce5d22
    </ex:ResourceId>
  </s12:Header>
  <s12:Body>
    <wsrp:QueryResourceProperties>
      <wsrp:QueryExpression
        dialect="http://www.w3.org/TR/1999/REC-xpath-19991116" >
        boolean(/*/NumberOfBlocks > 20 and */BlockSize=1024)
      </wsrp:QueryExpression>
    </wsrp:QueryResourceProperties>
  </s12:Body>
</s12:Envelope>
```

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

```
<s12:Envelope
  xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
```



```

xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
xmlns:wsrp=
"http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
xmlns:resp="http://www.other.org/otherNS">
<s12:Header>
  <wsa:Action>
    http://www.ibm.com/xmlns/stdwip/web-services/WS-
ResourceProperties/QueryResourcePropertiesResponse
  </wsa:Action>
  <wsa:To s12:mustUnderstand="1">
    http://www.requestor.org/someEndpoint
  </wsa:To>
  <resp:SomeResourceId>
    uuid:9fef5fec-6dc3-44a2-ba32-8680cace43f9
  </resp:SomeResourceId>
</s12:Header>
<s12:Body>
  <wsrp:QueryResourcePropertiesResponse>
    true
  </wsrp:QueryResourcePropertiesResponse>
</s12:Body>
</s12:Envelope>

```

6 Subscription

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

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

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

WS-ResourceProperties defines the Notification Topic and TopicSpace elements [WS-Topics] that MUST be used to express the organization of the WS-Resource property element value change notifications. By understanding the relationship between Topics and resource properties, and examining the set of Topics supported by the NotificationProducer Web service, the service requestor can determine which of the resource properties are able to participate in the value change notification pattern.

The Topic and TopicSpace elements associated with resource property value change notification are described as follows:

1. The WS-Resource's resource properties document MAY be defined using resource properties declared in multiple XML namespaces. For each of these XML namespaces, an associated TopicSpace element MUST be defined. The TopicSpace element defines a topic space intended to contain topics related to value changes of resource properties declared in that XML namespace.
 - The value of the TopicSpace element's targetNamespace attribute MUST be the same as the URI of the namespace in which the resource property element is defined. The name attribute of the TopicSpace element SHOULD have the value "ResourcePropertiesTopicSpace".
2. For each resource property participating in the value change notification pattern, a Topic element MUST be defined as a child of the TopicSpace element defined in 1.
 - Notification messages reflecting changes to the resource property are associated with this Topic.
 - The value of the Topic element's name attribute MUST be the same as the NCName of the resource property element.
 - The value of the Topic element's messageTypes attribute MUST include wsrp:ResourcePropertyValueChangeNotification (defined later in this section). In addition, it MAY include QNames of other message elements.
 - A designer MAY introduce additional child sub-topic elements to the topic element that represent application specific needs.
3. The WS-Resource acting as the NotificationProducer MUST include Topics defined in 2, as part of the value of its "Topics" resource property element. One such Topic MUST be included for each resource property element offered as a target for a value change subscription.
4. When a WS-Resource observes a resource property value change, it SHOULD create a notification message that expresses the situation, and associate the notification message with the Topic associated with that resource property. Note: there are many circumstances in which a change to a resource property does not result in the generation of a notification message. For example, a resource property value may change frequently, making generation of notification messages too expensive for the service. In this situation, a WS-Resource may choose to never generate notification message artifacts to record value change, or it may choose to generate notification message artifacts for a subset of the value change situations.

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

```
<wsrp:ResourcePropertyValueChangeNotification>  
  <wsrp:OldValue> xsd:any </wsrp:OldValue?>  
  <wsrp:NewValue> xsd:any </wsrp:NewValue>  
</wsrp:ResourcePropertyValueChangeNotification>
```

This element may appear as the root element of the notification message, or it may appear as a descendent of the root, accommodating patterns where the notification

message itself is contained in an enveloping mechanism. The form of the ResourcePropertyValueChangeNotification is further constrained as follows:

/wsrp:ResourcePropertyValueChangeNotification

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

/wsrp:ResourcePropertyValueChangeNotification/OldValue

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

/wsrp:ResourcePropertyValueChangeNotification/NewValue

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

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.

8 Security Considerations

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

8.1 *Securing the message exchanges*

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

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

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

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

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

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

- **Message alteration** – Alteration is prevented by including signatures of the message information using WS-Security.
- **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-Security.
- **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing secured policies – see WS-Policy [WS-Policy] and WS-SecurityPolicy [WS-SecurityPolicy]).
- **Authentication** – Authentication is established using the mechanisms described in WS-Security and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- **Accountability** – Accountability is a function of the type of and string of the key and algorithms being used. In many cases, a strong symmetric key provides sufficient accountability. However, in some environments, strong PKI signatures are required.
- **Availability** – Many services are subject to a variety of availability attacks. Replay is a common attack and it is RECOMMENDED that this be addressed as described in the next bullet. Other attacks, such as network-level denial of service attacks are harder to avoid and are outside the scope of this specification. That said, care should be taken to ensure that minimal processing be performed prior to any authenticating sequences.
- **Replay** – Messages may be replayed for a variety of reasons. To detect and eliminate this attack, mechanisms should be used to identify replayed messages such as the timestamp/nonce outlined in WS-Security and the sequences outlined in WS-ReliableMessaging [WS-ReliableMessaging].

8.2 Securing Resource Properties

Since WS-ResourceProperties defines a mechanism to expose properties of a WS-Resource, security policies should be established that ensure that only authorized requestors can access the value of a resource property. In order to secure access to the resource properties, the message exchanges that provide the access should be appropriately controlled. Authorization policies should be put in place so that the implications of providing the state information (through GetResourceProperty, GetMultipleResourceProperties, or QueryResourceProperties messages) or through notification of value change and modification of the resource properties, are taken into account. These policies should take into account the semantic difference between components of the SetResourceProperties message i.e., that an update component is to update a value of a resource property whereas insert and delete components actually modify whether the WS-Resource contains the resource property values themselves.

The authorization policies may also reflect the sensitivity of the resource property(ies) that are accessible from a WS-Resource. The policies can be at a coarse granularity of the message exchange (e.g., Get(Multiple)ResourceProperty(ies) vs SetResourceProperty) itself. There would be many scenarios where finer grained control to individual resource properties may be desired (e.g., user Bob can access value of "Manufacturer" but not

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

Even if the requestor is authorized to access the requested resource properties, it is RECOMMENDED that the resource properties that are exchanged between a requestor and Web service are secured to ensure integrity and/or confidentiality of the resource property values. This will prevent unauthorized alternation and/or access to the property values while in transit. This would mean that the specific resource property elements are signed and/or encrypted within the message by leveraging WS-Security as discussed in the previous section.

9 Acknowledgements

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

This specification has been developed as a result of joint work with many individuals and teams. The authors wish to acknowledge the contributions from many people, including:

Tim Banks, Nick Butler, Glen Daniels, Andrew Eisenberg, Tom Freund, Diane Jordan, Jarek Gawor, Heather Kreger, Simon Laws, Susan Malaika, David Martin, Andreas Meier, Bryan Murray, Peter Niblett, Ian Robinson, John Rofrano, Ellen Stokes and Jay Unger.

10 References

[OGSI 1.0]

Open Grid Services Infrastructure (OGSI) V1.0

<http://forge.gridforum.org/projects/ggf-editor/document/draft-ogsi-service-1/en/1>

[SOAP 1.2]

<http://www.w3.org/TR/soap12-part1/>

[State Paper]

<http://www-106.ibm.com/developerworks/webservices/library/ws-resource/ws-modelingresources.pdf>

[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.

[WS-Addressing]

<http://www.ibm.com/developerworks/webservices/library/ws-add/>

[WS-AtomicTransaction]

<http://www.ibm.com/developerworks/webservices/library/ws-atomtran/>

[WS-BaseNotification]

<ftp://www6.software.ibm.com/software/developer/library/ws-notification/WS-BaseN.pdf>

[WS-Notification]

<http://www-106.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf>

[WS-Policy]

<http://www.ibm.com/developerworks/library/ws-policy>

[WS-ReliableMessaging]

<http://www.ibm.com/developerworks/webservices/library/ws-rm/>

[WS-ResourceLifetime]

<http://www-106.ibm.com/developerworks/webservices/library/ws-resource/ws-resourcelifetime.pdf>

[WS-SecureConversation]

<http://www.ibm.com/developerworks/library/ws-secon/>

[WS-Security]

<http://www.oasis-open.org/committees/download.php/5531/oasis-200401-wss-soap-message-security-1.0.pdf>

[WS-SecurityPolicy]

<http://www.ibm.com/developerworks/library/ws-secpol/>

[WS-Topics]

<ftp://www6.software.ibm.com/software/developer/library/ws-notification/WS-Topics.pdf>

[WS-Trust]

<http://www.ibm.com/developerworks/library/ws-trust/>

[WSDL 2.0]

<http://www.w3.org/TR/wsd12/>

[XML-Infoset]

<http://www.w3.org/TR/xml-infoset/>

[XPATh]

<http://www.w3.org/TR/xpath>

11 Appendix I – XML Schema

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Legal Disclaimer
```

Copyright Notice

(c) Copyright Computer Associates International, Inc.,
Fujitsu Limited, Hewlett-Packard Development Company,
International Business Machines Corporation and
The University of Chicago 2003, 2004. All Rights Reserved.

```
-->
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsrp=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  targetNamespace=
    "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
>

<!-- ===== Global Attribute Declaration for WSDL 1.1 portType===== -->
<xsd:attribute name="ResourceProperties" type="xsd:QName" />

<!-- = Notification Message for ResourceProperties value change === -->
<xsd:complexType name="ResourcePropertyValueChangeNotificationType">
  <xsd:sequence>
    <xsd:element name="OldValue" nillable="true"
      minOccurs="0" maxOccurs="1" >
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any minOccurs="1" maxOccurs="1" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="NewValue" nillable="true"
      minOccurs="1" maxOccurs="1" >
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any minOccurs="1" maxOccurs="1" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="ResourcePropertyValueChangeNotification"
  type="wsrp:ResourcePropertyValueChangeNotificationType" />

<xsd:complexType name="QueryExpressionType" mixed="true">
  <xsd:sequence>
    <xsd:any minOccurs="0" maxOccurs="1" processContents="lax" />
  </xsd:sequence>
  <xsd:attribute name="dialect" type="xsd:anyURI" />
</xsd:complexType>

<xsd:element name="QueryExpression" type="wsrp:QueryExpressionType"
/>

</xsd:schema>
```


12 Appendix II – WSDL 1.1

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

```
<?xml version="1.0" encoding="utf-8"?>
<!--
    Legal Disclaimer

    Copyright Notice

    (c) Copyright Computer Associates International, Inc.,
        Fujitsu Limited, Hewlett-Packard Development Company,
        International Business Machines Corporation and
        The University of Chicago 2003, 2004. All Rights Reserved.
-->

<wsdl:definitions name="WS-ResourceProperties"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:wsrp=
        "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
    targetNamespace=
        "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties"
>

<!-- ===== Types Definitions ===== -->
  <wsdl:types>
    <xsd:schema
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace=
        "http://www.ibm.com/xmlns/stdwip/web-services/WS-ResourceProperties">

      <xsd:include schemaLocation=
        "http://www-106.ibm.com/developerworks/webservices/library/ws-
        resource/WS-ResourceProperties.xsd"
      />

      <xsd:import
        namespace=
          "http://schemas.xmlsoap.org/ws/2003/03/addressing"
        schemaLocation=
          "http://schemas.xmlsoap.org/ws/2003/03/addressing"
      />

<!-- ===== Common fault information to carry in all fault messages =====
-->
    <xsd:complexType name="BaseFaultType">
      <xsd:sequence>
        <xsd:element name="Timestamp" type="xsd:dateTime"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="Originator"
          type="wsa:EndpointReferenceType">
```

```

        minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ErrorCode"
        minOccurs="0" maxOccurs="1">
        <xsd:complexType>
            <xsd:complexContent mixed="true">
                <xsd:extension base="xsd:anyType">
                    <xsd:attribute name="dialect"
type="xsd:anyURI"
                                use="required"/>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Description" type="xsd:string"
        minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="FaultCause" type="wsrp:BaseFaultType"
        minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

<!-- ===== Message Types for GetResourceProperty ===== -->

    <xsd:element name="GetResourcePropertyRequest"
        type="xsd:QName" />

    <xsd:element name="GetResourcePropertyResponse" >
        <xsd:complexType>
            <xsd:sequence>
                <xsd:any minOccurs="0" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:complexType name="ResourceUnknownFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsrp:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="ResourceUnknownFault"
        type="wsrp:ResourceUnknownFaultType"/>

    <xsd:complexType name="InvalidResourcePropertyQNameFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsrp:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="InvalidResourcePropertyQNameFault"
        type="wsrp:InvalidResourcePropertyQNameFaultType"/>

<!-- ===== Message Types for GetMultipleResourceProperties ===== -->
    <xsd:element name="GetMultipleResourcePropertiesRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ResourceProperty" type="xsd:QName"
                    minOccurs="1" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

```

```

    </xsd:complexType>
  </xsd:element>

  <xsd:element name="GetMultipleResourcePropertiesResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:any minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

<!-- ===== Message Types for SetResourceProperties ===== -->

  <xsd:complexType name="SetRequestComponentType" >
</xsd:complexType>

  <xsd:element name="SetRequestComponent"
    type="wsrp:SetRequestComponentType"
    abstract="true" />

  <xsd:complexType name="InsertType">
    <xsd:complexContent>
      <xsd:extension base="wsrp:SetRequestComponentType">
        <xsd:sequence>
          <xsd:any processContents="lax"
            minOccurs="1" maxOccurs="1" />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="Insert"
    type="wsrp:InsertType"
    substitutionGroup="wsrp:SetRequestComponent" />

  <xsd:complexType name="UpdateType">
    <xsd:complexContent>
      <xsd:extension base="wsrp:SetRequestComponentType">
        <xsd:sequence>
          <xsd:any minOccurs="1" maxOccurs="1"
            processContents="lax" />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="Update"
    type="wsrp:UpdateType"
    substitutionGroup="wsrp:SetRequestComponent" />

  <xsd:complexType name="DeleteType">
    <xsd:complexContent>
      <xsd:extension base="wsrp:SetRequestComponentType">
        <xsd:attribute name="ResourceProperty"
          type="xsd:QName"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="Delete"

```

```

        type="wsrp:DeleteType"
        substitutionGroup="wsrp:SetRequestComponent" />

    <xsd:element name="SetResourceProperties">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="wsrp:SetRequestComponent"
                    minOccurs="1" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="SetResourcePropertiesResponse" >
        <xsd:complexType />
    </xsd:element>

    <xsd:complexType
name="InvalidSetResourcePropertiesRequestContentFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsrp:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element
name="InvalidSetResourcePropertiesRequestContentFault"
type="wsrp:InvalidSetResourcePropertiesRequestContentFaultType"/>

    <xsd:complexType name="UnableToModifyResourcePropertyFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsrp:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="UnableToModifyResourcePropertyFault"
type="wsrp:UnableToModifyResourcePropertyFaultType"/>

    <xsd:complexType
name="SetResourcePropertyRequestFailedFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsrp:BaseFaultType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="SetResourcePropertyRequestFailedFault"
type="wsrp:SetResourcePropertyRequestFailedFaultType"/>

<!-- ===== Message Types for QueryResourceProperties ===== -->

    <xsd:element name="QueryResourcePropertiesRequest" >
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="wsrp:QueryExpression"
                    minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
        </xsd:complexType>

```

```

    </xsd:element>

    <xsd:element name="QueryResourcePropertiesResponse" >
      <xsd:complexType>
        <xsd:complexContent mixed="true">
          <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
              <xsd:any processContents="lax"
                minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>

    <xsd:complexType name="UnknownQueryExpressionDialectFaultType">
      <xsd:complexContent>
        <xsd:extension base="wsrp:BaseFaultType"/>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="UnknownQueryExpressionDialectFault"
      type="wsrp:UnknownQueryExpressionDialectFaultType"/>

    <xsd:complexType name="InvalidQueryExpressionFaultType">
      <xsd:complexContent>
        <xsd:extension base="wsrp:BaseFaultType"/>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="InvalidQueryExpressionFault"
      type="wsrp:InvalidQueryExpressionFaultType"/>

    <xsd:complexType name="QueryEvaluationErrorFaultType">
      <xsd:complexContent>
        <xsd:extension base="wsrp:BaseFaultType"/>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="QueryEvaluationErrorFault"
      type="wsrp:QueryEvaluationErrorFaultType"/>

    </xsd:schema>
  </wsdl:types>

  <!-- ===== GetResourceProperty =====
  GetResourceProperty(QName)
  returns: any
-->
  <wsdl:message name="GetResourcePropertyRequest">
    <wsdl:part name="GetResourcePropertyRequest"
      element="wsrp:GetResourcePropertyRequest" />
  </wsdl:message>

  <wsdl:message name="GetResourcePropertyResponse">
    <wsdl:part name="GetResourcePropertyResponse"
      element="wsrp:GetResourcePropertyResponse" />
  </wsdl:message>

```

```

<wsdl:message name="ResourceUnknownFault">
  <part name="ResourceUnknownFault"
    element="wsrp:ResourceUnknownFault" />
</wsdl:message>

<wsdl:message name="InvalidResourcePropertyQNameFault">
  <part name="InvalidResourcePropertyQNameFault"
    element="wsrp:InvalidResourcePropertyQNameFault" />
</wsdl:message>

<!-- =====GetMultipleResourceProperties =====
GetMultipleResourceProperties(list of QName)
returns: sequence of any
-->
<wsdl:message name="GetMultipleResourcePropertiesRequest">
  <wsdl:part name="GetMultipleResourcePropertiesRequest"
    element="wsrp:GetMultipleResourcePropertiesRequest" />
</wsdl:message>

<wsdl:message name="GetMultipleResourcePropertiesResponse">
  <wsdl:part name="GetMultipleResourcePropertiesResponse"
    element="wsrp:GetMultipleResourcePropertiesResponse" />
</wsdl:message>

<!-- ===== SetResourceProperties =====
SetResourceProperties(
{ insert (any) |
  update@QName (any) |
  delete@QName } +
)
returns: empty
-->
<wsdl:message name="SetResourcePropertiesRequest">
  <wsdl:part name="SetResourcePropertiesRequest"
    element="wsrp:SetResourceProperties" />
</wsdl:message>

<wsdl:message name="SetResourcePropertiesResponse">
</wsdl:message>

<wsdl:message name="InvalidSetResourcePropertiesRequestContentFault">
  <part name="InvalidSetResourcePropertiesRequestContentFault"
element="wsrp:InvalidSetResourcePropertiesRequestContentFault" />
</wsdl:message>

<wsdl:message name="UnableToModifyResourcePropertyFault">
  <part name="UnableToModifyResourcePropertyFault"
    element="wsrp:UnableToModifyResourcePropertyFault" />
</wsdl:message>

<wsdl:message name="SetResourcePropertyRequestFailedFault">
  <part name="SetResourcePropertyRequestFailedFault"
    element="wsrp:SetResourcePropertyRequestFailedFault" />
</wsdl:message>

<!-- ===== QueryResourceProperties =====

```

```

QueryResourceProperties (QueryExpression)
returns: any
-->
<wsdl:message name="QueryResourcePropertiesRequest">
  <wsdl:part name="QueryResourcePropertiesRequest"
    element="wsrp:QueryResourcePropertiesRequest" />
</wsdl:message>

<wsdl:message name="QueryResourcePropertiesResponse">
  <wsdl:part name="QueryResourcePropertiesResponse"
    element="wsrp:QueryResourcePropertiesResponse" />
</wsdl:message>

<wsdl:message name="UnknownQueryExpressionDialectFault">
  <part name="UnknownQueryExpressionDialectFault"
    element="wsrp:UnknownQueryExpressionDialectFault" />
</wsdl:message>

<wsdl:message name="InvalidQueryExpressionFault">
  <part name="InvalidQueryExpressionFault"
    element="wsrp:InvalidQueryExpressionFault" />
</wsdl:message>

<wsdl:message name="QueryEvaluationErrorFault">
  <part name="QueryEvaluationErrorFault"
    element="wsrp:QueryEvaluationErrorFault" />
</wsdl:message>

<!-- ===== PortType Definitions ===== -->
<wsdl:portType name="GetResourceProperty">
  <wsdl:operation name="GetResourceProperty">
    <wsdl:input name="GetResourcePropertyRequest"
      message="wsrp:GetResourcePropertyRequest" />
    <wsdl:output name="GetResourcePropertyResponse"
      message="wsrp:GetResourcePropertyResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrp:ResourceUnknownFault" />
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
      message="wsrp:InvalidResourcePropertyQNameFault" />
  </wsdl:operation>
</wsdl:portType>

<wsdl:portType name="GetMultipleResourceProperties">
  <wsdl:operation name="GetMultipleResourceProperties">
    <wsdl:input name="GetMultipleResourcePropertiesRequest"
      message="wsrp:GetMultipleResourcePropertiesRequest"
    />
    <wsdl:output name="GetMultipleResourcePropertiesResponse"
      message="wsrp:GetMultipleResourcePropertiesResponse"
    />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrp:ResourceUnknownFault" />
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
      message="wsrp:InvalidResourcePropertyQNameFault" />
  </wsdl:operation>
</wsdl:portType>

```

```

<wsdl:portType name="SetResourceProperties">
  <wsdl:operation name="SetResourceProperties">
    <wsdl:input name="SetResourcePropertiesRequest"
      message="wsrp:SetResourcePropertiesRequest" />
    <wsdl:output name="SetResourcePropertiesResponse"
      message="wsrp:SetResourcePropertiesResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrp:ResourceUnknownFault" />
    <wsdl:fault
name="InvalidSetResourcePropertiesRequestContentFault"
message="wsrp:InvalidSetResourcePropertiesRequestContentFault" />
    <wsdl:fault name="UnableToModifyResourcePropertyFault"
      message="wsrp:UnableToModifyResourcePropertyFault"
/>
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
      message="wsrp:InvalidResourcePropertyQNameFault" />
    <wsdl:fault name="SetResourcePropertyRequestFailedFault"
      message="wsrp:SetResourcePropertyRequestFailedFault"
/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:portType name="QueryResourceProperties">
  <wsdl:operation name="QueryResourceProperties">
    <wsdl:input name="QueryResourcePropertiesRequest"
      message="wsrp:QueryResourcePropertiesRequest" />
    <wsdl:output name="QueryResourcePropertiesResponse"
      message="wsrp:QueryResourcePropertiesResponse" />
    <wsdl:fault name="ResourceUnknownFault"
      message="wsrp:ResourceUnknownFault" />
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
      message="wsrp:InvalidResourcePropertyQNameFault" />
    <wsdl:fault name="UnknownQueryExpressionDialectFault"
      message="wsrp:UnknownQueryExpressionDialectFault" />
    <wsdl:fault name="InvalidQueryExpressionFault"
      message="wsrp:InvalidQueryExpressionFault" />
    <wsdl:fault name="QueryEvaluationErrorFault"
      message="wsrp:QueryEvaluationErrorFault" />
  </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```