Web Services Resource 1.2 (WS-Resource)

Working Draft 02, 9 December 2004

Document identifier:

wsrf-WS-Resource-1.2-draft-02

Location:

http://docs.oasis-open.org/wsrf/2004/11/wsrf-WS-Resources-1.2-draft-02.pdf

Editors:

Steve Graham, IBM <sggraham@us.ibm.com>
Anish Karmarkar, Oracle <Anish.Karmarkar@oracle.com>
Jeff Mischkinsky, Oracle <jeff.mischkinsky@oracle.com>
Ian Robinson, IBM <ian_robinson@uk.ibm.com>
Igor Sedukhin, Computer Associates <Igor.Sedukhin@ca.com>

Abstract:

This specification defines a WS-Resource, which describes the relationship between a Web service and a resource in the WS-Resource Framework. This document also defines the term WS-Resource Access Pattern, the abstract concept of how resources are accessed through Web services, as well as several concrete embodiments based on various Web services referencing mechanisms.

Status:

This document is published by this TC as a "working draft". It is possible that it 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-subscribe@lists.oasis-open.org list. To subscribe, send an email message to wsrf-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to

the Intellectual Property Rights section of the WSRF TC web page (http://www.oasis-open.org/committees/wsrf/).

Table of Contents

1 Introduction

This specification defines a WS-Resource, which describes the relationship between a Web service and a resource in the WS-Resource Framework. This document also defines the term WS-Resource Access Pattern, the abstract concept of how resources are accessed through Web services, as well as several concrete embodiments based on various Web services referencing mechanisms.

1.1 Goals and Requirements

The goal of WS-Resource is to standardize the terminology and concepts needed to express the relationship between Web services and resources.

1.1.1 Requirements

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

- Define the term “resource”
- Define the term “WS-Resource”, describing the relationship between Web services and resources.
- Define the term “WS-Resource Access Pattern”, the abstract means by which a resource can be distinguished in a message exchange between a requestor and a Web service.
- Define one or more concrete embodiments of the WS-Resource Access Pattern.

1.2 Terminology

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]).

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

- `?′ denotes optionality (i.e. zero or one occurrences),
- `*′ denotes zero or more occurrences,
- `+′ one or more occurrences,
- `[′ and `]′ are used to form groups,
- `|′ represents choice.
- Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema.

```
<!-- sample pseudo-schema -->
<element
    required_attribute_of_type_QName="xs:QName"
    optional_attribute_of_type_string="xs:string"? >
  <required_element />
  <optional_element />?
```

```
  <one_or_more_of_these_elements />+
  [ <choice_1 /> | <choice_2 /> ]*
</element>
```

1.3 Namespaces

The following namespaces are used in this document:

Prefix	Namespace
s12	http://www.w3.org/2003/05/soap-envelope
xs	http://www.w3.org/2001/XMLSchema
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing
wsdl	http://schemas.xmlsoap.org/wsdl
wsrfmd	http://docs.oasis-open.org/wsrf/2004/10/ws-rap/ws-md.xsd

2 WS-Resource Terminology

The following terms are important to define the relationship between a Web service and one or more resources.

2.1 Resource

A resource is a logical entity that has the following characteristics:

- It MUST be identifiable; a resource has at least one resource identifier (see Section 2.2).
- It MUST have a set of zero or more properties, which are expressible in XML infoset.
- It MAY have lifecycle.

2.2 Resource Identifier

A resource identifier embodies sufficient information required to distinguish one resource from all other resources within its scope of identification.

2.3 WS-Resource

A WS-Resource is a Web service through which a resource can be accessed. A WS-Resource is further defined as follows:

- An identifier of the resource MUST appear as part of any message to a WS-Resource to allow the WS-Resource to disambiguate the resource targeted by the message. We refer to this pattern of access as the "**WS-Resource Access Pattern**".
- The set of properties of the resource MUST be expressed using an XML Infoset described by XML schema. The WS-Resource MUST support accessing resource properties through message exchanges defined by the WS-Resource Properties specification [WSRF-RP].
- If access to the lifecycle of the resource is exposed through the WS-Resource, the WS-Resource MAY support the message exchanges defined by the WS-Resource Lifetime specification [WSRF-RL].

Note: there are circumstances under which the resource identifier of the resource also appears as application data in the message. A message which otherwise satisfies the WS-Resource Access Pattern, and in which a resource identifier *also* appears in the message does not violate the WS-Resource Access Pattern.

2.4 WS-Resource Reference

A WS-Resource reference (or just reference) is a representation through which a single WS-Resource can be accessed. A reference encapsulates a resource identifier and may contain other information necessary to access the WS-Resource.

For a given resource identifier there may be many references. The way two references are compared for equality is implementation specific and not defined by this specification.

3 WS-Resource Access Pattern Embodiments

As defined above, the term “WS-Resource Access Pattern” defines a concept describing how a Web service disambiguates which resource is targeted by a message to a WS-Resource. There are many ways in which this can be achieved. We refer to a concrete realization of the WS-Resource Access Pattern as an “embodiment”. A WS-Resource MUST support at least one embodiment. A message exchange conformant to the WS-Resource Access Pattern is NOT required to implement all embodiments of the WS-Resource Access Pattern.

Each embodiment of the WS-Resource Access Pattern MUST:

- Specify the form of the WS-Resource reference
- Specify how the resource identifier appears in the WS-Resource reference
- Specify how a resource identifier appears in the message

Each embodiment SHOULD provide a non-normative, simple XML example illustrating how the embodiment achieves the requirements of being a WS-Resource Access Pattern embodiment.

The following sections define an initial set of embodiments of the WS-Resource Access Pattern. Applications may define additional embodiments.

3.1 WS-Addressing

This embodiment is one in which WS-Addressing is used [WSA].

In this embodiment, the form of the reference to a WS-Resource is an endpoint reference, or more precisely an XML element whose type is, or is derived (by extension) from the complexType named EndpointReferenceType defined by the WS-Addressing specification.

The address of the Web service endpoint part of the WS-Resource is contained in the wsa:Address element information item of the endpoint reference. There are two ways in which the resource identifier may appear:

1)in the contents of the wsa:ReferenceProperty element information item of the endpoint reference (Note, the wsa:ReferenceProperty element information item MUST have at least one child element information item)

or

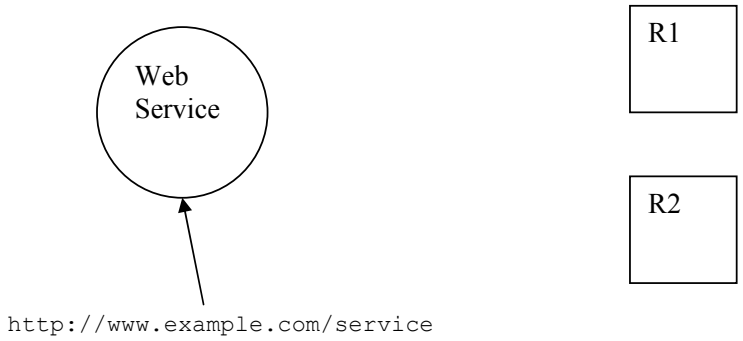
2) embedded as part of the wsa:Address element information item of the endpoint reference.

We label (non-normatively) the first style of encoding the resource identifier encoding as “WS-Addressing embodiment using Reference Properties” and we label (non-normatively) the second style of encoding the resource identifier as “WS-Addressing embodiment using Address”.

In a message that is conformant to this embodiment of the WS-Resource Access Pattern, the address of the Web service endpoint and the resource identifier of the resource must appear in the message according to binding specific rules outlined in WS-Addressing. For example, in the SOAP binding defined by WS-Addressing, the Web service endpoint address is contained in the wsa:Address element information item in the endpoint reference and appears in the message as the contents of the wsa:To SOAP header and each direct child element information item (if any) of the wsa:ReferenceProperties element information item appears in the message as separate SOAP headers.

3.1.1 Example

The following diagram illustrates and example set of components that comprise a small collection of WS-Resources:



In the example above, there is one Web service that has a URL address of "http://www.example.com/service". This Web service provides access to two resources, identified simply as "R1" and "R2". A reference to the WS-Resource associated with this Web service and the resource identified by "R1" would appear as follows:

```
<wsa:EndpointReference>
   <wsa:Address>http://www.example.com/service</wsa:Address>
   <wsa:ReferenceProperties>
      <tns:SomeDisambiguatorElement>R1</tns:SomeDisambiguatorElement>
   </wsa:ReferenceProperties> ?
…
</wsa:EndpointReference>
```

This reference uses the form of this embodiment labeled as "WS-Addressing embodiment using Reference Properties". An example GetResourceProperties message, in a SOAP/HTTP binding, following this embodiment of the WS-Resource Access Pattern would look as follows:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
            xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
            xmlns:wsrf-rp="http://docs.oasis-open.org/wsrf/2004/11/wsrf-WS-
ResourceProperties-1.2-draft-05.xsd">
  <S:Header>
       <wsa:To> http://www.example.com/service </wsa:To>
       <wsa:Action>
    http://docs.oasis-open.org/wsrf/2004/11/WS-ResourceProperties/GetResourceProperty
       </wsa:Action>
       <tns:SomeDisambiguatorElement>R1</tns:SomeDisambiguatorElement>
    ...
  </S:Header>
  <S:Body>
```

```
      <wsrf-rp:GetResourceProperty …
    ...
  </S:Body>
</S:Envelope>
```

3.2 WSDL 1.1 Service Element Embodiment

This embodiment is one in which WSDL 1.1 is used [WSDL11]. The form of a reference is a WSDL definitions element which contains exactly one WSDL service child element which, in turn, contains one or more WSDL port child elements each bound to the same portType element. Each port offers a potentially different binding to the same WS-Resource,

The resource identifier MUST be encoded within the child element(s) of the port element that specify the address as defined by WSDL 1.1. In case of SOAP binding, within the soap:address element.

In this embodiment, the address contained within the WSDL port element contains both the address of the Web service endpoint and the resource identifier.

For example, the following is a valid reference to a WS-Resource in this embodiment:

```
<wsdl:definitions … >
  <wsdl:service name="svc">
    <wsdl:port … >
      <soap:address="http://www.example.com/R1"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

In this case, messages sent to http://www.example.com/R1 are, actually, sent to the endpoint of the Web service which provides access to the resource in this example identified by the string "R1". Note that even though resource identifier does not appear within the SOAP envelope contained in messages associated with this reference, it MUST appear in as part of the HTTP message (in the form of the URL).

3.3 WSDL 1.1 Binding Element Embodiment

Comment: This embodiment is a subtle one and requires further explanation and debate.

This embodiment is one in which WSDL 1.1 is used [WSDL11]. The form of a reference to a WS-Resource is a WSDL definitions element which contains exactly one WSDL service child element which, in turn, contains one or more WSDL port child elements each bound to the same portType element.

The information about the resource identifier is encoded within the child element(s) of the port element and the child element(s) of the binding element to which the port element refers to (binding attribute). The information in the WSDL binding element and port element, or any extension thereof, describes to the requestor how to form messages to a WS-Resource that contain the resource identifier. In order for the requestor to properly format a message to a WS-Resource, it MUST understand the rules of the WSDL binding element and port element and the meaning of any contained extension elements (see http://www.w3.org/TR/wsdl#_ports, http://www.w3.org/TR/wsdl#_bindings).

Note that a form of a reference to a WS-Resource in this embodiment of WS-RAP does not necessarily contain a value of the resource identifier, however it 1) identifies what the resource

identifier is (e.g. the fact that it is an X.509 certificate in the caller's context), and 2) specifies where in the message the value of the resource identifier has to appear (e.g. a <soap:header> WSDL SOAP binding extension element). The actual value of the resource identifier depends on the application specifics and the context in which the requestor/caller runs. Therefore, it is possible to have one WS-Resource reference which when interpreted in each requestor/caller context will result in messages targeted to different resources. This is an application-context dependant form of a reference.

For example, WSDL SOAP binding may specify a custom header as follows.

```
<wsdl:definitions … xmlns:tns="…" xmlns:my="…">
  <wsdl:message name="custom">
    <wsdl:part name="hdr" element="my:ResourceIdentifier"/>
  </wsdl:message>
  <wsdl:binding name="SOAP" … >
    <wsdl:operation … >
      <wsdl:input>
        <soap:header message="tns:custom" part="hdr" use="literal"/>
        <soap:body … > … </soap:body>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="svc">
    <wsdl:port name="SOAPHTTP" binding="tns:SOAP">
      <soap:address="http://my.server.org/soap/http/listener"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

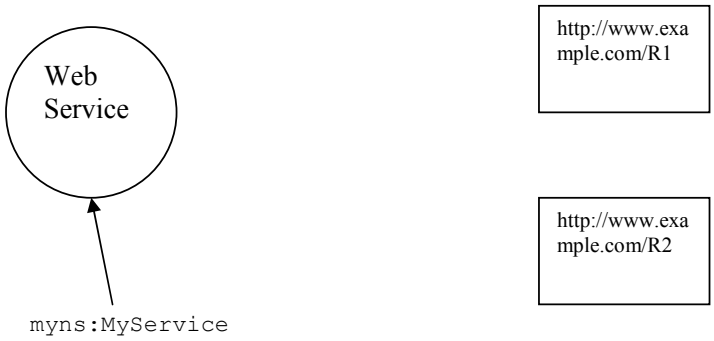
In this example, the requestor would need to understand how to form the contents of the **<my:ResourceIdentifier>** element before sending a SOAP message to the **http://my.server.org/soap/http/listener** address. The QName of this header element identifies the application semantics of the element. Precisely how to form the contents of the required header element is the application semantics, and it has to be known and implemented by the requestor. This specification does not make any assumptions as to what such application semantics could be.

3.4 WS-MessageDelivery Embodiment

This embodiment is based on WS-MessageDelivery Version 1.0 [WSMD]. This embodiment defines the form of the reference to a WS-Resource, namely wsrfmd:WSResourceReference, and a normative dereferencing mechanism when using the SOAP protocol.

3.4.1 Example

The following diagram illustrates an example set of components that comprise two WS-Resources:



In the example above, there is one Web service that is identified by the WSDL service QName "myns:MyService". This Web service provides access to two resources, identified as "http://www.example.com/R1" and "http://www.example.com/R2". A reference to the WS-Resource associated with this Web service and the resource identified by "http://www.example.com/R1" would appear as follows:

```
<wsrfmd:WSResourceReference>
  <!-- Web service reference -->
  <wsrfmd:WSReference wsmd:wsdlLocation="http://example.com/wsdlloc">
    <wsmd:serviceQName xmlns:myns="http://example.com/myns">
      myns:MyService
    </wsmd:serviceQName>
  </wsrfmd:WSReference>
  <!—resource identifier -->
  <wsrfmd:ResourceIdentifier uri="http://www.example.com/R1" />
</wsrfmd:WSResourceReference>
```

The reference to the WS-Resource consists of the QName of the WSDL service element that identifies the Web service and the URI [URI] "http://www.example.com/R1" -- the resource identifier.

An example GetResourceProperties message, when using SOAP, following this embodiment of the WS-Resource Access Pattern would look as follows:

```
<S:Envelope>
  <S:Header>
    <wsmd:MessageDestination>...</wsmd:MessageDestination>
    <wsmd:MessageOriginator>...</wsmd:MessageOriginator>
    <wsmd:OperationName>...<wsmd:OperationName>
    <wsrfmd:ResourceIdentifier uri="http://www.example.com/R1" />
  </S:Header>
  <S:Body>...</S:Body>
</S:Envelope>
```

The value of the resource identifier is sent as a separate SOAP header block.

3.4.2 WSResourceReference

In this embodiment, the form of the reference to a WS-Resource is wsrfmd:WSResourceReference, or more precisely an element information item whose type is, or is derived from, wsrfmd:WSResourceReferenceType as defined in Appendix B. The following pseudo-schema describes the contents of this element:

```
<wsrfmd:WSResourceReference>
    <wsrfmd:WSReference>wsmd:destination</wsrfmd:WSReference>
    <wsrfmd:ResourceIdentifier uri="xs:anyURI"?>
        any
    </wsrfmd:ResourceIdentifier>?
</wsrfmd:WSResourceReference>
```

wsrfmd:WSResourceReference element information item contains a reference to a Web service (either a WSDL service element or a QName that identifies a WSDL service element) and an optional resource identifier as defined in Section 3.4.3.

The element information item wsrfmd:WSReference MUST conform to WS-MessageDelivery Version 1.0. This requires that the WSDL service element MUST conform to section 2.1 of [WSMD]. The wsrfmd:WSReference element information item identifies the Web service to which messages targeted for the WS-Resource are sent.

The element wsrfmd:ResourceIdentifier, if present, specifies the identity of the resources. If the element wsrfmd:ResourceIdentifier is absent then the resource is identified by the WSDL service element itself.

3.4.3 ResourceIdentifier

This element information item identifies the resource and is specified by the following pseudo-schema:

```
<wsrfmd:ResourceIdentifier uri="xs:anyURI"?>
```

```
    any
</wsrfmd:ResourceIdentifier>
```

The entire wsrfmd:ResourceIdentifier information element represents the resource identifier in this embodiment.

This element is part of the WS-Resource reference as well as a SOAP header block as defined in Section 3.4.4. When used as a SOAP header block, all the SOAP processing rules related to SOAP header blocks apply.

3.4.4 Dereferencing WSResourceReference using SOAP

When a messages is targeted to a particular WS-Resource, the entire wsrfmd:ResourceIdentifier information element, if present, is included in the message in a protocol/binding specific way. This section defines this mapping when using SOAP. It is expected that mappings for other protocols/bindings will be defined by other specifications.

To dereference and send a message to a WS-Resource identified by wsrfmd:WSResourceReference using SOAP:

1. The Web service to which the message to be sent is identified by the contents of wsrfmd:WSReference – this either contains a WSDL service element or a QName that identifies the WSDL service element. A port that supports a SOAP binding within that service element is selected.
2. When accessing an operation on the selected port by sending a message to the WS-Resource, wsrfmd:ResourceIdentifier element, if present in the WSResourceReference MUST be sent as a SOAP header block.

The content of the SOAP header block, if present, identifies the resource targeted by the message. When dereferencing a WSResourceReference the message exchange MUST conform to the WSDL and WS-MessageDelivery specifications.

4 References

4.1 Normative

[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.

[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.

[WSA] http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/

[WSDL 1.1] http://www.w3.org/TR/wsdl

[WSMD] http://www.w3.org/Submission/2004/SUBM-ws-messagedelivery-20040426

[WS-ResourceLifetime] http://docs.oasis-open.org/wsrf/2004/11/wsrf-WS-ResourceLifetime-1.2-draft-04.pdf

[WS-ResourceProperties] http://docs.oasis-open.org/wsrf/2004/11/wsrf-WS-ResourceProperties-1.2-draft-05.pdf

[XML-Infoset] http://www.w3.org/TR/xml-infoset/

4.2 Non-Normative

[SOAP 1.2] http://www.w3.org/TR/soap12-part1/

[WSDL 2.0] http://www.w3.org/TR/wsdl12/

Appendix A. Acknowledgments

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

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

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

Appendix B. XML Schema for WS-MessageDelivery Embodiment

The XML Schema types and element used by the WS-MessageDelivery embodiment are defined in the following XML schema:

```
<?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.

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:wsmd="http://www.w3.org/2004/04/ws-messagedelivery"
    xmlns:wsrfmd="http://docs.oasis-open.org/wsrf/2004/10/ws-rap/ws-
md.xsd"

    targetNamespace=" http://docs.oasis-open.org/wsrf/2004/10/ws-
rap/ws-md.xsd"
    elementFormDefault="qualified">

  <xs:import namespace="http://www.w3.org/2004/04/ws-messagedelivery"/>

  <!-- holder for resource identifier -->
  <xs:element name="ResourceIdentifier"
      type="wsrfmd:ResourceIdentifierType"/>
  <xs:complexType name="ResourceIdentifierType" >
    <xs:sequence>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="uri" type="xs:anyURI" />
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

  <!-- syntactic struct that contains the reference to the WS and the
       resource identifier -->
  <xs:element name="WSResourceReference"
```

```
      type="wsrfmd:WSResourceReferenceType"/>
  <xs:complexType name="WSResourceReferenceType">
    <xs:sequence>
      <xs:element name="WSReference" type="wsmd:destination"/>
      <xs:element ref="wsrfmd:ResourceIdentifier" minOccurs="0"/>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

</xs:schema>
```

Appendix C. Revision History

Rev	Date	By Whom	What
wd-01	2004-08-27	Steve Graham	Initial version created based on 08/23 and 08/24 meeting amongst the authors.
wd-02	2004-09-02	sgg	Modifications per feedback on 09/01 telecon, and email from Anish and Igor.
wd-01.a-f	Various	sgg	Reflected various progress
wd-01g	2004-09-29	sgg	Reflected final agreements
wd-02a	2004-10-07	ir	Editorial and TC issues
Wd-02.b	2004-11-22	sgg	Resolved WSRF75 and WSRF76
Wd-02	2004-12-09	ir	Editorial

Appendix D. Notices

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.