Web Services Addressing - WSDL Binding

W3C Working Draft 8 December 2004

This version:
http://www.w3.org/TR/2004/WD-ws-addr-wsdl-20041208
Latest version:
http://www.w3.org/TR/ws-addr-wsdl
Previous versions:

Editors:
Martin Gudgin, Microsoft Corp
Marc Hadley, Sun Microsystems, Inc

This document is also available in these non-normative formats: postscript, PDF, XML, and plain text.

>

Abstract<a>

Web Services Addressing provides transport-neutral mechanisms to address Web services and messages. Web Services Addressing WSDL Binding (this document) defines how the abstract properties defined in Web Services Addressing Core are described using WSDL.

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This is the First Public Working Draft of the Web Services Addressing - WSDL Binding specification for review by W3C members and other interested parties. It has been produced by the Web Services Addressing Working Group (WG), which is part of the W3C Web Services Activity.

In this Working Draft, the Web Services Addressing Working Group has, in keeping with its charter, separated the WS-Addressing Member Submission into three separate specifications: Core, SOAP Binding, and WSDL Binding. The Working Group expects to publish an updated draft in the near future incorporating more resolutions from its issues list.

Discussion of this document takes place on the public public public-ws-addressing@w3.org mailing list (public archive). Comments on this specification should be sent to this mailing list.

This document was produced under the 5 February 2004 W3C Patent Policy. The Working Group maintains a public list of patent disclosures relevant to this document; that page also includes instructions for disclosing [and excluding] a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) with respect to this specification should disclose the information in accordance with section 6 of the W3C Patent Policy.

Per section 4 of the W3C Patent Policy, Working Group participants have 150 days from the title page date of this document to exclude essential claims from the W3C RF licensing requirements with respect to this document series. Exclusions are with respect to the exclusion reference document, defined by the W3C Patent Policy to be the latest version of a document in this series that is published no later than 90 days after the title page date of this document.

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Editorial note	
The Web Services Addressing Working Group has decided to use XML Schema, where appropriate, to describe constructs defined in this specification. Note that this restricts use of Web Services Addressing to XML 1.0.	

>

Short Table of Contents<a>

>

Table of Contents<a>

Appendices

>

>1. Introduction

Web Services Addressing Core*WS-Addressing-Core [p.8]* defines a set of abstract properties and an XML Infoset [*XML Information Set [p.8]*] representation thereof to identify Web service endpoints and to secure end-to-end identification of endpoints in messages. Web Services Addressing WSDL Binding (this document) defines how the abstract properties defined in Web Services Addressing Core are described using WSDL.

1.1 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 [*IETF RFC 2119 [p.8]*].

When describing abstract data models, this specification uses the notational convention used by the XML Infoset [*XML Information Set [p.8]*]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [*XML Schema Structures [p.8]* , *XML Schema Datatypes [p.8]*], this specification uses the notational convention of WS-Security [*WS-Security [p.9]*]. 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 (<xs:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

1.2 Namespaces

This specification uses a number of namespace prefixes throughout; they are listed in Table 1-1 [p.4] . Note that the choice of any namespace prefix is arbitrary and not semantically significant (see [*XML Namespaces [p.8]*]).

Table 1-1. Prefixes and Namespaces used in this specification

Prefix	Namespace
S	http://www.w3.org/2003/05/soap-envelope
S11	http://schemas.xmlsoap.org/soap/envelope
wsa	http://www.w3.org/2004/12/addressing
xs	http://www.w3.org/2001/XMLSchema

WS-Addressing is defined in terms of the XML Information Set [*XML Information Set [p.8]*]. WS-Addressing is designed to be able work with WSDL 2.0 [*WSDL 2.0 [p.8]*] and also (for backwards compatibility with WSDL 1.1 [*WSDL 1.1 [p.9]*] described services. WS-Addressing may be used with SOAP [*SOAP 1.2 Part 1: Messaging Framework [p.9]*] as described in Web Services Addressing - SOAP Binding[*WS-Addressing-SOAP [p.8]*]. The examples in this specification use an XML 1.0 [*XML 1.0 [p.8]*] representation but this is not a requirement.

All information items defined by WS-Addressing are identified by the XML namespace URI [*XML Namespaces [p.8]*] "http://www.w3.org/2004/12/addressing". A normative XML Schema [*XML Schema Structures [p.8] , XML Schema Datatypes [p.8]*] document can be obtained by dereferencing the XML namespace URI.

2. Endpoint References

This specification introduces a new description element type, the endpoint reference, with the intent of supporting a set of dynamic usage patterns not currently appropriately covered by WSDL 1.1 [*WSDL 1.1 [p.9]*].

To support these scenarios, we define a lightweight and extensible mechanism to dynamically identify and describe service endpoints and instances. Because of the current limits of the WSDL 1.1 extensibility model, the WSDL 1.1 service and port elements cannot be used to cover the use cases listed above. Endpoint references logically extend the WSDL description model (e.g., portTypes, bindings, etc.), but do not replace it. Endpoint references will be used instead of WSDL <service/> elements in the following cases:

- Specific instances of a stateful service need to be identified or its instance-specific configuration details need to be transmitted.

- A lightweight, self-contained description of a service endpoint needs to be communicated. In particular, this may be necessary when the details of the endpoint configuration are already shared by the communicating parties, but specific policy information needs to be added or updated, typically as a result of a dynamic configuration process.

Endpoint references complement and do not replace the WSDL/1.1 <wsdl:service> element. We expect solutions built on WSDL/1.1 to continue to utilize a service element. Moving forward we anticipate that endpoint references and WSDL will evolve coherently. The endpoint references may link to service elements in WSDL/1.1, and support additional scenarios in which the WSDL information is not known by a party processing a message. These scenarios may include dynamic messaging or limited capability message processors.

2.1 Information Model for Endpoint References

The WSDL binding of Web Services Addressing introduces the following additional abstract properties:

[selected port type] : QName (0..1)

> The QName of the primary portType of the endpoint being conveyed.

[service-port] : (QName, NCName (0..1)) (0..1)

> This is the QName identifying the WSDL service element that contains the definition of the endpoint being conveyed. The service name provides a link to a full description of the service endpoint. An optional non-qualified name identifies the specific port in the service that corresponds to the endpoint.

3. Associating Action with WSDL Operations

WS-Addressing defines two mechanisms to associate an action with input, output and fault elements within a WSDL port type.

3.1 Explicit Association

The action may be explicitly associated using the wsa:Action attribute or in the absence of the attribute the action is defined by the rule in section 3.3.2.

For example consider the following WSDL excerpt:

Example 3-1. Explicit specification of wsa:Action value in a WSDL description.

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetTradePricesInput"
            wsa:Action="http://example.com/GetQuote"/>
      <output message="tns:GetTradePricesOutput"
            wsa:Action="http://example.com/Quote"/>
    </operation>
  </portType>
  ...
</definitions>
```

The action for the input of the GetLastTradePrice operation within the StockQuotePortType is explicitly defined to be http://example.com/GetQuote. The action for the output of this same operation is http://example.com/Quote.

3.2 Default Action Pattern

In the absence of the wsa:Action attribute, the following pattern is used to construct a default action for inputs and outputs. The general form of an action URI is as follows:

Example 3-2. Structure of defaulted wsa:Action URI.

```
[target namespace]/[port type name]/[input|output name]
```

The "/" is a literal character to be included in the action. The values of the properties are as defined below.

[target namespace] is the target namespace (/definition/@targetNamespace). If [target namespace] ends with a "/" an additional "/" is not added.

[port type name] is the name of the port type (/definition/portType/@name).

[input|output name] is the name of the element as defined in Section 2.4.5 of WSDL 1.1.

For fault messages, this pattern is not applied. Instead, the following URI is the default action URI for fault messages: `http://www.w3.org/2004/12/addressing/fault`

For example consider the following WSDL excerpt:

Example 3-3. Example WSDL without explicit wsa:Action values with explicit message names.

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetTradePricesInput" name="GetQuote"/>
      <output message="tns:GetTradePricesOutput" name="Quote"/>
    </operation>
  </portType>
  ...
</definitions>
```

[targetNamespace] = http://example.com/stockquote

[port type name] = StockQuotePortType

[input name] = GetQuote

[output name] = Quote

Applying the pattern above with these values we have:

input action = http://example.com/stockquote/StockQuotePortType/GetQuote

output action = http://example.com/stockquote/StockQuotePortType/Quote

WSDL defines rules for a default input or output name if the name attribute is not present. Consider the following example:

Example 3-4. Example WSDL without explicit wsa:Action values or explicit message names.

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetTradePricesInput"/>
      <output message="tns:GetTradePricesOutput"/>
    </operation>
  </portType>
  ...
</definitions>
```

[targetNamespace] = http://example.com/stockquote

[port type name] = StockQuotePortType

According to the rules defined in WSDL 2.4.5, if the name attribute is absent for the input of a request response operation the default value is the name of the operation "Request" appended.

[input name] = GetLastTradePriceRequest

Likewise, the output defaults to the operation name with "Response" appended.

[output name] = GetLastTradePriceResponse

Applying the pattern above with these values we have:

input action = http://example.com/stockquote/StockQuotePortType/GetLastTradePriceRequest

output action = http://example.com/stockquote/StockQuotePortType/GetLastTradePriceResponse

4. WS-Addressing and WSDL Message Exchange Patterns

TBD

4.1 WSDL 1.1 Message Exchange Patterns

TBD

4.2 WSDL 2.0 Message Exchange Patterns

TBD

5. References

[WS-Addressing-Core]
Web Services Addressing - Core, M. Gudgin, M. Hadley, Editors.

[WS-Addressing-SOAP]
Web Services Addressing - SOAP Binding, M. Gudgin, M. Hadley, Editors.

[WSDL 2.0]
Web Services Description Language 2.0, TBD.

[IETF RFC 2119]
Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, Author. Internet Engineering Task Force, June 1999. Available at http://www.ietf.org/rfc/rfc2119.txt.

[RFC 2396bis]
T. Berners-Lee, et al, "Uniform Resource Identifier (URI): Generic Syntax,", W3C/MIT, July 2004. (See *http://www.ietf.org/internet-drafts/draft-fielding-uri-rfc2396bis-07.txt*.)

[XML 1.0]
Extensible Markup Language (XML) 1.0 (Second Edition), T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. World Wide Web Consortium, 10 February 1998, revised 6 October 2000. This version of the XML 1.0 Recommendation is http://www.w3.org/TR/2000/REC-xml-20001006. The latest version of XML 1.0 is available at http://www.w3.org/TR/REC-xml.

[XML Namespaces]
Namespaces in XML, T. Bray, D. Hollander, and A. Layman, Editors. World Wide Web Consortium, 14 January 1999. This version of the XML Information Set Recommendation is http://www.w3.org/TR/1999/REC-xml-names-19990114. The latest version of Namespaces in XML is available at http://www.w3.org/TR/REC-xml-names.

[XML Information Set]
XML Information Set, J. Cowan and R. Tobin, Editors. World Wide Web Consortium, 24 October 2001. This version of the XML Information Set Recommendation is http://www.w3.org/TR/2001/REC-xml-infoset-20011024. The latest version of XML Information Set is available at http://www.w3.org/TR/xml-infoset.

[XML Schema Structures]
XML Schema Part 1: Structures, H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. World Wide Web Consortium, 2 May 2001. This version of the XML Schema Part 1 Recommendation is http://www.w3.org/TR/2001/REC-xmlschema-1-20010502. The latest version of XML Schema Part 1 is available at http://www.w3.org/TR/xmlschema-1.

[XML Schema Datatypes]
XML Schema Part 2: Datatypes, P. Byron and A. Malhotra, Editors. World Wide Web Consortium, 2 May 2001. This version of the XML Schema Part 2 Recommendation is http://www.w3.org/TR/2001/REC-xmlschema-2-20010502. The latest version of XML Schema Part 2 is available at http://www.w3.org/TR/xmlschema-2.

[SOAP 1.2 Part 1: Messaging Framework]
SOAP Version 1.2 Part 1: Messaging Framework, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. Frystyk Nielsen, Editors. World Wide Web Consortium, 24 June 2003. This version of the "SOAP Version 1.2 Part 1: Messaging Framework" Recommendation is http://www.w3.org/TR/2003/REC-soap12-part1-20030624/. The latest version of "SOAP Version 1.2 Part 1: Messaging Framework" is available at http://www.w3.org/TR/soap12-part1/.

[WSDL 1.1]
E. Christensen, et al, *Web Services Description Language (WSDL) 1.1*, March 2001.

[WS-Security]
OASIS, *Web Services Security: SOAP Message Security*, March 2004.

A. Acknowledgements (Non-Normative)

TBD

B. Change log (Non-Normative)

Date	Editor	Description
2004-11-23 @ 21:38	mhadley	Updated titles of examples. Fixed table formatting and references. Replaced uuid URIs with http URIs in examples. Added document status.
2004-11-11 @ 18:31	mgudgin	Added some TBD sections
2004-11-07 @ 02:03	mhadley	Second more detailed run through to separate core, SOAP and WSDL document contents. Removed dependency on WS-Policy. Removed references to WS-Trust and WS-SecurityPolicy
2004-11-02 @ 21:45	mhadley	Replaced hardcoded change log with one generated dynamically from CVS
2004-10-28 @ 18:09	mhadley	Fixed typo in abstract
2004-10-28 @ 17:05	mhadley	Initial cut of separating specification into core, soap and wsdl