Web Services Addressing - SOAP Binding

W3C Working Draft 8 December 2004

This version:
    http://www.w3.org/TR/2004/WD-ws-addr-soap-20041208
Latest version:
    http://www.w3.org/TR/ws-addr-soap
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 SOAP Binding (this document) defines the binding of the abstract properties defined in Web Services Addressing Core to SOAP Messages.

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 - SOAP 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 incorporporting 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.10]* defines a set of abstract properties and an XML Infoset [*XML Information Set [p.11]*] representation thereof to identify Web service endpoints and to secure end-to-end identification of endpoints in messages. Web Services Addressing SOAP Binding (this document) defines the binding of the abstract properties defined in Web Services Addressing Core to SOAP Messages.

The following example illustrates the use of these mechanisms in a SOAP 1.2 message being sent from http://business456.example/client1 to http://fabrikam123.example/Purchasing:

Example 1-1. Use of message addressing properties in a SOAP 1.2 message.

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
                xmlns:wsa="http://www.w3.org/2004/12/addressing"
         <wsa:MessageID>
(004)      http://example.com/someuniquestring
(005)    </wsa:MessageID>
(006)    <wsa:ReplyTo>
(007)      <wsa:Address>http://business456.example/client1</wsa:Address>
(008)    </wsa:ReplyTo>
(009)    <wsa:To>http://fabrikam123.example/Purchasing</wsa:To>
(010)    <wsa:Action>http://fabrikam123.example/SubmitPO</wsa:Action>
(011)   </S:Header>
(012)   <S:Body>
(013)     ...
(014)   </S:Body>
(015) </S:Envelope>
```

Lines (002) to (011) represent the header of the SOAP message where the mechanisms defined in the specification are used. The body is represented by lines (012) to (014).

Lines (003) to (010) contain the message information header blocks. Specifically, lines (003) to (005) specify the identifier for this message and lines (006) to (008) specify the endpoint to which replies to this message should be sent as an Endpoint Reference. Line (009) specifies the address URI of the ultimate receiver of this message. Line (010) specifies an Action URI identifying expected semantics.

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.10]*].

When describing abstract data models, this specification uses the notational convention used by XML Infoset [*XML Information Set [p.11]*]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [*XML Schema Structures [p.11]* , *XML Schema Datatypes [p.11]*], this specification uses the notational convention of WS-Security [*WS-Security [p.11]*]. 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.10]*]).

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.11]*]. WS-Addressing is conformant to the SOAP 1.2 [*SOAP 1.2 Part 1: Messaging Framework [p.11]*] processing model and is also compatible with SOAP 1.1[*SOAP 1.1 [p.11]*] for backwards compatibility. WS-Addressing may be used with WSDL [*WSDL 2.0 [p.10]*] described services as described in Web Services Addressing - WSDL Binding[*WS-Addressing-WSDL [p.10]*]. The examples in this specification use an XML 1.0 [*XML 1.0 [p.10]*] representation but this is not a requirement.

All information items defined by WS-Addressing are identified by the XML namespace URI [*XML Namespaces [p.10]*] "http://www.w3.org/2004/12/addressing". A normative XML Schema [*XML Schema Structures [p.11]* , *XML Schema Datatypes [p.11]*] document can be obtained by dereferencing the XML namespace URI.

2. Binding Endpoint References

This section defines the binding of Endpoint references to SOAP messages.

When a message needs to be addressed to the endpoint, the information contained in the endpoint reference is mapped to the message according to a transformation that is dependent on the protocol and data representation used to send the message. Protocol-specific mappings (or bindings) will define how the information in the endpoint reference is copied to message and protocol fields. This specification defines the SOAP binding for endpoint references. This mapping MAY be explicitly replaced by other bindings (defined as WSDL bindings or as policies); however, in the absence of an applicable policy stating that a different mapping must be used, the SOAP binding defined here is assumed to apply. To ensure interoperability with a broad range of devices, all conformant implementations MUST support the SOAP binding.

The SOAP binding for endpoint references is defined by the following two rules:

- The [address] property in the endpoint reference is copied in the [destination] message information property. The infoset representation of the [destination] property becomes a header block in the SOAP message.
- Each [reference property] and [reference parameter] element becomes a header block in the SOAP message. The element information item of each [reference property] or [reference parameter] (including all of its [children], [attributes] and [in-scope namespaces]) is to be added as a header block in the new message.

The next example shows how the default SOAP binding for endpoint references is used to construct a message addressed to the endpoint:

Example 2-1. Example endpoint reference.

```
<wsa:EndpointReference xmlns:wsa="..." xmlns:fabrikam="...">
   <wsa:Address>http://www.fabrikam123.example/acct</wsa:Address>
   <wsa:ReferenceProperties>
       <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
   </wsa:ReferenceProperties>
   <wsa:ReferenceParameters>
       <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
   </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

According to the mapping rules stated above, the address value is copied in the "To" header and the "CustomerKey" element should be copied literally as a header in a SOAP message addressed to this endpoint. The SOAP message would look as follows:

Example 2-2. Example endpoint reference mapped to SOAP message header blocks.

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
         xmlns:wsa="..." xmlns:fabrikam="... ">
   <S:Header>
     ...
    <wsa:To>http://www.fabrikam123.example/acct</wsa:To>
    <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
    <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
     ...
   </S:Header>
   <S:Body>
     ...
   </S:Body>
</S:Envelope>
```

3. Faults

The faults defined in this section are generated if the condition stated in the preamble in each subsection is met. They are sent to the [fault endpoint], if present and valid. Otherwise they are sent to the [reply endpoint] if present. If neither is present faults may be sent to the [source endpoint].

Endpoints compliant with this specification MUST include required message information headers on all fault messages. Fault messages are correlated as replies using the [relationship] property as defined in Section 3. The [action] property below designates WS-Addressing fault messages (this URI is also used as the default Action value for WSDL fault messages, as described in Section 3.3.2):

```
http://www.w3.org/2004/12/addressing/fault
```

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

[Detail] The detail element. If absent, no detail element is defined for the fault.

The properties above bind to a SOAP 1.2 fault as follows:

Example 3-1. Binding of fault properties to SOAP 1.2 messages.

```
<S:Envelope>
 <S:Header>
   <wsa:Action>
     http://www.w3.org/2004/12/addressing/fault
   </wsa:Action>
   <!-- Headers elided for clarity.  -->
 </S:Header>
 <S:Body>
  <S:Fault>
```

```
    <S:Code>
     <S:Value>[Code]</S:Value>
      <S:Subcode>
     <S:Value>[Subcode]</S:Value>
      </S:Subcode>
    </S:Code>
    <S:Reason>
      <S:Text xml:lang="en">[Reason]</S:Text>
    </S:Reason>
    <S:Detail>
      [Detail]
   </S:Detail>
   </S:Fault>
 </S:Body>
</S:Envelope>
```

The SOAP 1.1 fault is less expressive and map only [Subcode] and [Reason]. These the properties bind to a SOAP 1.1 fault as follows:

Example 3-2. Binding of fault properties to SOAP 1.1 messages.

```
<S11:Envelope>
 <S11:Body>
  <S11:Fault>
   <faultcode>[Subcode]</faultcode>
   <faultstring xml:lang="en">[Reason]</faultstring>
  </S11:Fault>
 </S11:Body>
</S11:Envelope>
```

3.1 Invalid Message Information Header

A message information header cannot be processed.

[Code] S:Sender

[Subcode] wsa:InvalidMessageInformationHeader

[Reason] A message information header is not valid and the message cannot be processed. The validity failure can be either structural or semantic, e.g. a [destination] that is not a URI or a [relationship] to a [message id] that was never issued.

[Detail] [invalid header]

3.2 Message Information Header Required

A required message information header is absent.

[Code] S:Sender

[Subcode] wsa:MessageInformationHeaderRequired

[Reason] A required message information header, To, MessageID, or Action, is not present.

[Detail] [Missing Header QName]

3.3 Destination Unreachable

No endpoint can be found capable of acting in the role of the [destination] property.

[Code] S:Sender

[Subcode] wsa:DestinationUnreachable

[Reason] No route can be determined to reach the destination role defined by the WS-Addressing To.

[Detail] empty

3.4 Action Not Supported

The [action] property in the message is not supported at this endpoint.

The contents of this fault are as follows:

[Code] S:Sender

[Subcode] wsa:ActionNotSupported

[Reason] The [action] cannot be processed at the receiver.

[Detail] [action]

3.5 Endpoint Unavailable

The endpoint is unable to process the message at this time either due to some transient issue or a permanent failure.

The endpoint may optionally include a RetryAfter parameter in the detail. The source should not retransmit the message until this duration has passed.

[Code] S:Receiver

[Subcode] wsa:EndpointUnavailable

[Reason] The endpoint is unable to process the message at this time.

[Detail] <wsa:RetryAfter ...>[xs:NonNegativeInteger]</wsa:RetryAfter>

The following describes the attributes and elements listed above:

/wsa:RetryAfter

This element (of type xs:NonNegativeInteger) is a suggested minimum duration in milliseconds to wait before retransmitting the message. If this element is omitted from the detail, the value is infinite.

/wsa:RetryAfter/@{any}

These optional extensibility attributes do not affect processing.

4. Security Considerations

It is strongly recommended that the communication between services be secured using the mechanisms described in WS-Security [*WS-Security [p.11]*]. In order to properly secure messages, the body and all relevant headers need to be included in the signature. Specifically, the message information headers described in this specification (e.g. <wsa:To>) need to be signed with the body in order to "bind" the two together. It should be noted that for messages traveling through intermediaries, it is possible that some or all of the message information headers may have multiple signatures when the message arrives at the ultimate receiver. It is strongly recommended that the initial sender include a signature to prevent any spoofing by intermediaries.

Whenever an address is specified (e.g. <wsa:From>, <wsa:ReplyTo>, <wsa:FaultTo>, ...), the processor should ensure that a signature is provided with claims allowing it to speak for the specified target in order to prevent certain classes of attacks (e.g. redirects). As well, care should be taken if the specified endpoint contains reference properties or parameters as unverified endpoint references could cause certain classes of header insertion attacks.

The message information headers blocks may have their contents encrypted in order to obtain end-to-end privacy, but care should be taken to ensure that intermediary processors have access to required information (e.g. <wsa:To>).

Some processors may use message identifiers (<wsa:MessageID>) as part of a uniqueness metric in order to detect replays of messages. Care should be taken to ensure that a unique identifier is actually used. For example, it may be appropriate in some scenarios to combine the message identifier with a timestamp.

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.
- Address spoofing – Address spoofing is prevented by ensuring that all address are signed by a party authorized to speak for (or on behalf of) the address.

- Key integrity – Key integrity is maintained by using the strongest algorithms possible (by comparing secured policies.
- Authentication – Authentication may be established using the mechanisms described in WS-Security.
- Accountability – Accountability is a function of the type of and strength 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 – All reliable messaging services are subject to a variety of availability attacks. Replay detection is a common attack and it is recommended that this be addressed by the mechanisms described in WS-Security and/or caching of message identifiers. 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 state is saved 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. Alternatively, and optionally, other technologies, such as sequencing, can also be used to prevent replay of application messages.

5. References

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

[WS-Addressing-WSDL]
Web Services Addressing - WSDL 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.

[SOAP 1.1]
Don Box, et al, *Simple Object Access Protocol (SOAP) 1.1*, May 2000.

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

A. Acknowledgements (Non-Normative)

TBD

B. Change log (Non-Normative)

Date	Editor	Description
2004-11-24 @ 15:32	mhadley	Added note that addressing is backwards compatible with SOAP 1.1
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.
2004-11-07 @ 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 @ 22:25	mhadley	Removed static change log and added dynamically generated change log from cvs.
2004-10-28 @ 17:05	mhadley	Initial cut of separating specification into core, soap and wsdl