Web Services Addressing 1.0 - SOAP Binding

W3C Proposed Recommendation 21 March 2006

This version:
http://www.w3.org/TR/2006/PR-ws-addr-soap-20060321
Latest version:
http://www.w3.org/TR/ws-addr-soap
Previous versions:
http://www.w3.org/TR/2005/CR-ws-addr-soap-20050817
Editors:
Martin Gudgin, Microsoft Corp
Marc Hadley, Sun Microsystems, Inc
Tony Rogers, Computer Associates International, Inc

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

Abstract

Web Services Addressing provides transport-neutral mechanisms to address Web services and messages. Web Services Addressing 1.0 - SOAP Binding (this document) defines the binding of the abstract properties defined in Web Services Addressing 1.0 - 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 Proposed Recommendation of the Web Services Addressing 1.0 - 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.

The publication of this document signifies a call for review by the W3C Advisory Committee. W3C Advisory Committee Representatives are invited to submit their formal review per the instructions in the Call for Review (see Advisory Committee questionnaires). The review period ends on 18 April 2006.

Members of the public are also invited to send comments on this Proposed Recommendation to the public public-ws-addressing-comments@w3.org mailing list (public archive).

This document addresses the comments received against the Candidate Recommendation draft previously published. The detailed disposition of those comments can be found in the Candidate Recommendation issues list. An implementation report showing that the Candidate Recommendation exit criteria have been met and exceeded is available, along with a test suite. A diff-marked version against the previous version of this document is available. For a detailed list of changes since the last publication of this document, please refer to appendix **B. Change Log** [p.23] .

The Working Group intends to maintain the value of the Web Services Addressing 1.0 - Core namespace URI that was assigned in the Candidate Recommendation unless significant changes are made that impact the implementation of the specification.

Discussion of this document takes place on the public-ws-addressing@w3.org mailing list (public archive). A list of formal objections against the set of WS-Addressing 1.0 Working Drafts is also available.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Publication as a Proposed Recommendation 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.

Table of Contents

Appendices

---

1. Introduction

Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*] defines a set of abstract properties and an XML Infoset [*XML Information Set [p.21]*] representation thereof to reference Web service endpoints and to facilitate end-to-end addressing of endpoints in messages. Web Services Addressing 1.0 - SOAP Binding (this document) defines the binding of the abstract properties defined in Web Services Addressing 1.0 - Core to SOAP Messages.

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

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

```
(01) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
               xmlns:wsa="http://www.w3.org/2005/08/addressing">
(02)   <S:Header>
(03)    <wsa:MessageID>http://example.com/6B29FC40-CA47-1067-B31D-00DD010662DA</wsa:MessageID>
(04)    <wsa:ReplyTo>
(05)      <wsa:Address>http://example.com/business/client1</wsa:Address>
(06)    </wsa:ReplyTo>
(07)    <wsa:To>http://example.com/fabrikam/Purchasing</wsa:To>
(08)    <wsa:Action>http://example.com/fabrikam/SubmitPO</wsa:Action>
(09)   </S:Header>
(10)   <S:Body>
(11)     ...
(12)   </S:Body>
(13) </S:Envelope>
```

Lines (02) to (09) represent the header of the SOAP message where the mechanisms defined in the specification are used. The body is represented by lines (10) to (12).

Lines (03) to (08) contain the message addressing properties serialized as SOAP header blocks. Specifically, line (03) specifies the identifier for this message and lines (04) to (06) specify the endpoint to which replies to this message should be sent as an Endpoint Reference. Line (07) specifies the address URI of the ultimate receiver of this message. Line (08) 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.21]*].

When describing abstract data models, this specification uses the notational convention used by XML Infoset [*XML Information Set [p.21]*]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [*XML Schema Structures [p.21]* , *XML Schema Datatypes [p.22]*], this specification uses the notational convention of WS-Security [*WS-Security [p.22]*]. 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.5] . Note that the choice of any namespace prefix is arbitrary and not semantically significant (see [*XML Namespaces [p.21]*]).

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/2005/08/addressing
wsaw	http://www.w3.org/2005/03/addressing/wsdl
xs	http://www.w3.org/2001/XMLSchema

WS-Addressing is defined in terms of the XML Information Set [*XML Information Set [p.21]*]. WS-Addressing is conformant to the SOAP 1.2 [*SOAP 1.2 Part 1: Messaging Framework [p.22]*] processing model and is also compatible with SOAP 1.1[*SOAP 1.1 [p.22]*] for backwards compatibility. WS-Addressing may be used with WSDL [*WSDL 2.0 [p.21]*] described services as described in Web Services Addressing 1.0 - WSDL Binding[*WS-Addressing-WSDL [p.21]*]. The examples in this specification use an XML 1.0 [*XML 1.0 [p.21]*] representation but this is not a requirement.

All information items defined by this specification are identified by the XML namespace URI [*XML Namespaces [p.21]*] http://www.w3.org/2005/08/addressing. A normative XML Schema [*XML Schema Structures [p.21]* , *XML Schema Datatypes [p.22]*] document can be obtained by dereferencing the XML namespace URI.

2. SOAP 1.2 Addressing 1.0 Feature

This section defines the SOAP 1.2 Addressing 1.0 Feature.

2.1 Feature Name

The SOAP 1.2 Addressing 1.0 Feature is named using the following URI:

- http://www.w3.org/2005/08/addressing/feature

2.2 Description

The SOAP 1.2 Addressing 1.0 Feature provides a SOAP-specific expression of the abstract message addressing properties defined by Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*].

This feature may be used with any SOAP MEP. A binding that supports this feature MUST provide a means to transmit the properties listed below with a message and to reconstitute their values on receipt of a message.

2.3 Properties

The SOAP 1.2 Addressing 1.0 Feature defines the following properties:

http://www.w3.org/2005/08/addressing/feature/Destination

Corresponds to the abstract [destination] property.

http://www.w3.org/2005/08/addressing/feature/SourceEndpoint

Corresponds to the abstract [source endpoint] property.

http://www.w3.org/2005/08/addressing/feature/ReplyEndpoint

Corresponds to the abstract [reply endpoint] property.

http://www.w3.org/2005/08/addressing/feature/FaultEndpoint

Corresponds to the abstract [fault endpoint] property.

http://www.w3.org/2005/08/addressing/feature/Action

Corresponds to the abstract [action] property.

http://www.w3.org/2005/08/addressing/feature/MessageID

Corresponds to the abstract [message id] property.

http://www.w3.org/2005/08/addressing/feature/Relationship

Corresponds to the abstract [relationship] property.

http://www.w3.org/2005/08/addressing/feature/ReferenceParameters

Corresponds to the abstract [reference parameters] property.

2.4 Interactions with Other SOAP Features

If the http://www.w3.org/2003/05/soap/features/action/Action property of the SOAP Action feature[*SOAP 1.2 Part 2: Adjuncts [p.22]*] has a value, then the value of the http://www.w3.org/2005/08/addressing/feature/Action property of the SOAP 1.2 Addressing 1.0 feature MUST be identical to it. Failure to have an identical value results in an Invalid Addressing Header fault (see **6.4.1 Invalid Addressing Header** [p.16]).

3. SOAP 1.2 Addressing 1.0 Module

The SOAP 1.2 Addressing 1.0 Module defines a set of SOAP header blocks to support the SOAP 1.2 Addressing 1.0 Feature described in **2. SOAP 1.2 Addressing 1.0 Feature** [p.5] .

3.1 Module Name

The SOAP 1.2 Addressing 1.0 Module is identified using the following URI:

- http://www.w3.org/2005/08/addressing/module

3.2 Description

The SOAP 1.2 Addressing 1.0 Feature (see **2. SOAP 1.2 Addressing 1.0 Feature** [p.5]) defines a set of SOAP properties and their correspondence to the abstract message addressing properties defined by Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*]. The SOAP 1.2 Addressing 1.0 Module defines SOAP headers corresponding to the XML Infoset representation of the abstract message addressing properties defined in Web Services Addressing 1.0 - Core.

3.2.1 Sending Messages

When sending a message each property is represented using the appropriate element information item as a SOAP header block. By default, the resulting header blocks are targeted at the ultimate recipient in the SOAP message path (note that extensions to WS-Addressing could be written to specify different targetting). **3.4 Binding Message Addressing Properties** [p.8] describes additional processing required when binding message addressing properties to SOAP header blocks.

3.2.2 Receiving Messages

When receiving a message, the abstract properties are populated from their corresponding element information items in the message. A message MUST NOT contain more than one wsa:To, wsa:ReplyTo, wsa:FaultTo, wsa:Action, or wsa:MessageID header targeted at a recipient; headers with an incorrect cardinality MUST NOT be used to populate the corresponding abstract properties. A recipient MUST generate a wsa:InvalidAddressingHeader (see **6.4.1 Invalid Addressing Header** [p.16]) fault if such a message is received.

Note:

The SOAP processing model dictates that message addressing properties targeted at an intermediary do not normally get relayed as message addressing properties when the message is forwarded along the message path. The specification for a SOAP header used as a reference property or use of the soap:relay attribute can override this default behavior.

3.3 Additional Infoset Items

The SOAP 1.2 Addressing 1.0 Module defines the following additional XML Infoset items:

/[reference parameters]/@wsa:IsReferenceParameter

> This REQUIRED attribute (of type xs:boolean) signifies whether the message addressing header is a reference parameter, see section **3.4 Binding Message Addressing Properties** [p.8] for more details on its use.

3.4 Binding Message Addressing Properties

When a message is to be addressed to an endpoint, the XML Infoset representation of each message addressing property that has been assigned a value is inserted into the message as a SOAP header block subject to the following additional constraints:

- The value, if any, of the [reference parameters] property is added to the SOAP message header: the element information item of each of the [reference parameters] (including all of its [children], [attributes] and [in-scope namespaces]) is added as a SOAP header block in the new message.

  Note:

  The insertion of SOAP headers into a message implies particular semantics. Since the reference parameter mechanism does not restrict the content of the generated headers, EPR suppliers should exercise appropriate caution to ensure their reference parameters do not cause unintended or erroneous semantics in the resultant SOAP message. For example, using a reference parameter to send a WS-Security[*WS-Security [p.22]*] header would be ill-advised (since other parts of the SOAP infrastructure will often control this header, and there must be at most one of them per message).

- Each header block added as a result of the above rule is annotated with a wsa:IsReferenceParameter attribute (see **3.3 Additional Infoset Items** [p.8]) whose value is a valid xs:boolean representation of "true". Any existing wsa:IsReferenceParameter attribute on the header block is replaced.

  Note:

  Integrity validation of [reference parameters] needs to take into account the addition of wsa:IsReferenceParameter attributes and the corresponding introduction of the WS-Addressing namespace to the [in-scope namespaces]

- The value of each message addressing property that is of type IRI MUST be serialized as an absolute IRI in the corresponding SOAP header block. No additional %-escaping is performed.
- Each optional element or attribute that has a value equal to the defined default value for that element or attribute MAY be omitted.

The following example shows how the SOAP 1.2 Addressing 1.0 Module is used to construct a message addressed to the endpoint:

Example 3-1. Example endpoint reference.

```
<wsa:EndpointReference
     xmlns:wsa="http://www.w3.org/2005/08/addressing"
     xmlns:wsaw="http://www.w3.org/2005/03/addressing/wsdl"
     xmlns:fabrikam="http://example.com/fabrikam"
     xmlns:wsdli="http://www.w3.org/2006/01/wsdl-instance"
     wsdli:wsdlLocation="http://example.com/fabrikam
       http://example.com/fabrikam/fabrikam.wsdl">
   <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
   <wsa:Metadata>
       <wsaw:InterfaceName>fabrikam:Inventory</wsaw:InterfaceName>
   </wsa:Metadata>
   <wsa:ReferenceParameters>
       <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
       <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
   </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

The address value is copied in the "To" header block and the "CustomerKey" and "ShoppingCart" elements are copied literally as a header blocks in a SOAP message addressed to this endpoint. The resulting SOAP message would look as follows:

Example 3-2. Example endpoint reference mapped to SOAP message header blocks.

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
         xmlns:wsa="http://www.w3.org/2005/08/addressing"
         xmlns:fabrikam="http://example.com/fabrikam">
   <S:Header>
     ...
    <wsa:To>http://example.com/fabrikam/acct</wsa:To>
    <wsa:Action>...</wsa:Action>
    <fabrikam:CustomerKey wsa:IsReferenceParameter='true'>123456789</fabrikam:CustomerKey>
    <fabrikam:ShoppingCart wsa:IsReferenceParameter='true'>ABCDEFG</fabrikam:ShoppingCart>
     ...
   </S:Header>
   <S:Body>
     ...
   </S:Body>
</S:Envelope>
```

3.5 Relationship between SOAP Headers and transport-level properties

Some underlying protocols may support native properties similar to the Message Addressing Properties. For example, the reply-to: email header is similar to the [reply endpoint] Message Addressing Property. Authors and implementors of bindings should not assume any particular correspondence between native properties and Message Addressing Properties. For example, if an email message represents only one hop in a multi-hop path, then the reply-to: header is likely to differ from the [reply endpoint] address.

4. SOAP 1.1 Addressing 1.0 Extension

The SOAP 1.1 Addressing 1.0 Extension defines a set of SOAP header blocks to support the SOAP 1.2 Addressing 1.0 Feature described in **2. SOAP 1.2 Addressing 1.0 Feature** [p.5] . This SOAP 1.1 extension is provided for backwards compatibility only.

4.1 Extension Name

The SOAP 1.1 Addressing 1.0 Extension is identified using the following URI:

- http://www.w3.org/2005/08/addressing/module

4.2 Description

The SOAP 1.2 Addressing 1.0 Feature (see **2. SOAP 1.2 Addressing 1.0 Feature** [p.5]) defines a set of SOAP properties and their correspondence to the abstract message addressing properties defined by Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*]. The SOAP 1.1 Addressing 1.0 Extension uses the XML Infoset representation of the abstract message addressing properties defined in Web Services Addressing 1.0 - Core and binds each element information item to a SOAP header block. The SOAP 1.1 Addressing 1.0 Extension operates as described in **3. SOAP 1.2 Addressing 1.0 Module** [p.7] with the following exceptions:

SOAP Action

> Use of the SOAPAction HTTP request header field is required when using the SOAP 1.1 HTTP binding. The field-value of the SOAPAction HTTP request header MUST either be the value of the [action] property enclosed in quotation marks, or the empty value "". The latter case supports the ability to obscure the [action] property through SOAP-level security mechanisms, without requiring otherwise unnecessary transport-level security. Any other value for SOAPAction results in an Invalid Message Addressing Property fault (see **6.4.1 Invalid Addressing Header** [p.16]).

5. Addresses in SOAP

In the following text, the term 'response endpoint' refers to the [reply endpoint] and [fault endpoint] message addressing properties collectively.

5.1 Use of Anonymous Address in SOAP Response Endpoints

A value of "http://www.w3.org/2005/08/addressing/anonymous" for the [destination] property implies no additional semantics beyond those resulting from the rules defined below and as described in Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*]. In particular, note that Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*], section 3.4 requires such a value in messages sent to a response endpoint whose [address] is "http://www.w3.org/2005/08/addressing/anonymous".

5.1.1 SOAP 1.1/HTTP

When "http://www.w3.org/2005/08/addressing/anonymous" is specified for the response endpoint then there is no change to the SOAP 1.1/ HTTP binding.

5.1.2 SOAP 1.2

When "http://www.w3.org/2005/08/addressing/anonymous" is specified for the response endpoint and the message is the http://www.w3.org/2003/05/soap/mep/InboundMessage property of a SOAP request-response MEP [*SOAP 1.2 Part 2: Adjuncts [p.22]*], then any response MUST be the http://www.w3.org/2003/05/soap/mep/OutboundMessage property of the same instance of the SOAP request-response MEP [*SOAP 1.2 Part 2: Adjuncts [p.22]*].

5.2 Use of Non-Anonymous Addresses in SOAP Response Endpoints

5.2.1 SOAP 1.1/HTTP

When "http://www.w3.org/2005/08/addressing/anonymous" is not specified for the response endpoint, then the message SHOULD be part of a binding that supports not returning a SOAP envelope in the HTTP response (e.g. see [*SOAP 1.1 Request Optional Response HTTP Binding [p.22]*]). Any response message SHOULD be sent using a separate connection and using the address value specified by response endpoint. Note that other specifications MAY define special URIs that have other behaviors (similar to the anonymous URI).

5.2.2 SOAP 1.2

When "http://www.w3.org/2005/08/addressing/anonymous" is not specified for the response endpoint, then any response SHOULD NOT be the http://www.w3.org/2003/05/soap/mep/OutboundMessage property of the same instance of the SOAP request-response MEP [*SOAP 1.2 Part 2: Adjuncts [p.22]*]. For instance, a SOAP 1.2 HTTP binding that supports a one-way MEP could put the reply message in a separate one-way MEP and a separate HTTP request. As in SOAP 1.1/HTTP, note that other specifications MAY define special URIs that have other behaviors (similar to the anonymous URI).

6. Faults

The faults defined in this section are generated if the condition stated in the preamble in each subsection is met.

Endpoints compliant with this specification MUST include the required message addressing properties serialized as SOAP headers in generated fault messages. Fault messages are correlated as replies using the [relationship] property as defined in Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*]. Note that omission of the [message id] property in an input message may impact the ability of a fault message receiver to correlate the fault message to the message that caused the fault message to be generated. Omission of the [fault endpoint] or [reply endpoint] properties in input messages may impact the delivery of a generated fault message

The [action] property below designates WS-Addressing fault messages:

`http://www.w3.org/2005/08/addressing/fault`

This action SHOULD NOT be used as an action value in messages other than those carrying WS-Addressing faults.

SOAP modules, extensions and applications SHOULD define custom [action] values for the faults they describe but MAY designate use of the following [action] value instead:

`http://www.w3.org/2005/08/addressing/soap/fault`

The above [action] value SHOULD be used for SOAP defined faults including version mismatch, must understand, and data encoding unknown.

Each of the predefined faults listed below is defined by specifying values for the following abstract properties:

[Code] The fault code, use of the specified fault code is REQUIRED.

[Subcode] The fault subcode, use of the specified fault subcode is REQUIRED.

[Subsubcode] A more specific fault subcode that may be used to further qualify the value of the [Subcode] property, use of a specified fault subcode is OPTIONAL.

[Reason] The English language reason element, use of the specified fault code is RECOMMENDED but alternate text MAY be used.

[Details] The detail elements, use of the specified detail elements is REQUIRED. If absent, no detail elements are defined for the fault.

6.1 SOAP 1.2 Fault Binding

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

[Code]

> The value of the [Code] property is bound as the value of the SOAP faults S:Fault/S:Code/S:Value element information item.

[Subcode]

The value of the [Subcode] property is bound as the value of the SOAP faults S:Fault/S:Code/S:Subcode/S:Value element information item.

[Subsubcode]

The value of the [Subsubcode] property is bound as the value of the SOAP faults S:Fault/S:Code/S:Subcode/S:/Subcode/S:Value element information item.

[Reason]

The value of the [Reason] property is bound as the value of the SOAP faults S:Fault/S:Reason/S:Text element information item.

[Details]

The value of the [Details] property is bound as child elements of the SOAP faults S:Fault/S:Detail element information item.

Example 6-1. Binding of fault properties to SOAP 1.2 messages.

```
<S:Envelope>
  <S:Header>
    <wsa:Action>http://www.w3.org/2005/08/addressing/fault</wsa:Action>
    <!-- Headers elided for brevity.  -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <S:Code>
        <S:Value>[Code]</S:Value>
        <S:Subcode>
          <S:Value>[Subcode]</S:Value>
          <S:Subcode>
            <S:Value>[Subsubcode]</S:Value>
          </S:Subcode>
        </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>
```

6.2 SOAP 1.1 Fault Binding

The SOAP 1.1 fault is slightly less expressive than the SOAP 1.2 fault and maps only [Subcode], [Reason] and [Detail]. These the properties bind to a SOAP 1.1 fault as follows:

[Subcode] or [Subsubcode]

> The value of the [Subsubcode] or, if that is not specified, the value of the [Subcode] property is bound as the value of the SOAP faults S11:Fault/faultcode element.

[Reason]

> The value of the [Reason] property is bound as the value of the SOAP faults S11:Fault/faultstring element.

[Details]

> The SOAP 1.1 fault detail is only for use with faults related to the body of a message and is therefore not used for SOAP 1.1 faults related to processing of addressing headers. Instead the value of the [Details] property is bound as the value of a new wsa:FaultDetail SOAP header block. The following describes the wsa:FaultDetail element:
>
> /wsa:FaultDetail
>
> > Zero or more of the elements defined in **6.3 Fault Detail Elements** [p.15] .
>
> /wsa:FaultDetail/@{any}
>
> > Optional extensibility attributes including SOAP role and mustUnderstand.

Example 6-2. Binding of fault properties to SOAP 1.1 messages.

```
<S11:Envelope>
  <S11:Header>
    <wsa:Action>http://www.w3.org/2005/08/addressing/fault</wsa:Action>
    <wsa:FaultDetail>[Details]</wsa:FaultDetail>
    <!-- Other headers elided for brevity.  -->
  </S11:Header>
  <S11:Body>
    <S11:Fault>
      <faultcode>[Subcode] or [Subsubcode]</faultcode>
      <faultstring xml:lang="en">[Reason]</faultstring>
    </S11:Fault>
  </S11:Body>
</S11:Envelope>
```

6.3 Fault Detail Elements

The following subsections define a set of elements used to convey additional information in the faults described in **6.4 Predefined Faults** [p.16] .

6.3.1 Problem Header QName

The following describes the <wsa:ProblemHeaderQName> element:

/wsa:ProblemHeaderQName

A QName representing the name of the root element of the problem header block.

/wsa:ProblemHeaderQName/@{any}

Optional extensibility attributes that do not affect processing.

6.3.2 Problem IRI

The following describes the <wsa:ProblemIRI> element:

/wsa:ProblemIRI

The IRI that caused the problem.

/wsa:ProblemIRI/@{any}

Optional extensibility attributes that do not affect processing.

6.3.3 Problem Action

The following describes the <wsa:ProblemAction> element:

/wsa:ProblemAction/wsa:Action

An optional element that provides the [action] that caused the problem.

/wsa:ProblemAction/wsa:SoapAction

An optional element that provides the SOAPAction IRI that caused the problem.

/wsa:ProblemAction/{any}

Optional extensibility elements that do not affect processing.

/wsa:ProblemAction/@{any}

Optional extensibility attributes that do not affect processing.

6.3.4 Retry After

The following describes the <wsa:RetryAfter> element:

/wsa:RetryAfter

This element (whose content is of type xs:unsignedLong) is a suggested minimum duration in milliseconds to wait before retransmitting the message. Omission of this element indicates that a retry is never likely to succeed.

/wsa:RetryAfter/@{any}

Optional extensibility attributes that do not affect processing.

6.4 Predefined Faults

6.4.1 Invalid Addressing Header

A header representing a WS-Addressing 1.0 Message Addressing Property is invalid and cannot be processed. The validity failure can be either structural or semantic, e.g. a [destination] that is not an IRI or a [relationship] to a [message id] that was never issued.

[Code] a QName representing the value S:Sender

[Subcode] a QName representing the value wsa:InvalidAddressingHeader

[Reason] the string: "A header representing a Message Addressing Property is not valid and the message cannot be processed"

[Details] either a <wsa:ProblemHeader> element that conveys a copy of the offending header or a <wsa:ProblemHeaderQName> element that conveys the QName of the root element of the offending header.

The invalid addressing header fault can be further narrowed in scope by use of the additional [Subsubcode]s specified in the following subsections. Use of these [Subsubcode] values is OPTIONAL.

6.4.1.1 wsa:InvalidAddress

Specifies that an [address] was invalid.

6.4.1.2 wsa:InvalidEPR

Specifies that the invalid header was expected to be an EPR but was not valid.

6.4.1.3 wsa:InvalidCardinality

Specifies that there was a greater than expected number of the specified header.

6.4.1.4 wsa:MissingAddressInEPR

Specifies that the invalid header was expected to be an EPR but did not contain an [address].

6.4.1.5 wsa:DuplicateMessageID

Specifies that the invalid header conveyed a [message id] that was a duplicate of one already received.

6.4.1.6 wsa:ActionMismatch

Specifies that the [action] and SOAPAction for the message did not match, [Details] MAY contain a <wsa:ProblemAction> element in addition to the <wsa:ProblemHeader> element or <wsa:ProblemHeaderQName> element.

6.4.1.7 wsa:OnlyAnonymousAddressSupported

Specifies that the only address supported is the anonymous address.

6.4.1.8 wsa:OnlyNonAnonymousAddressSupported

Specifies that the anonymous address is not supported, and that only a non-anonymous address will be accepted.

6.4.2 Message Addressing Header Required

A required header representing a Message Addressing Property is absent.

[Code] a QName representing the value S:Sender

[Subcode] a QName representing the value wsa:MessageAddressingHeaderRequired

[Reason] the string: "A required header representing a Message Addressing Property is not present"

[Details] a <wsa:ProblemHeaderQName> element that conveys the QName of the message addressing header that was missing.

6.4.3 Destination Unreachable

The endpoint identified by the value of [destination] property cannot be reached.

[Code] a QName representing the value S:Sender

[Subcode] a QName representing the value wsa:DestinationUnreachable

[Reason] the string: "No route can be determined to reach [destination]"

[Details] an optional <wsa:ProblemIRI> element that conveys the [address] of the [destination].

Implementation of this fault is optional.

6.4.4 Action Not Supported

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

[Code] a QName representing the value S:Sender

[Subcode] a QName representing the value wsa:ActionNotSupported

[Reason] the string: "The [action] cannot be processed at the receiver"

[Details] a <wsa:ProblemAction> element with a REQUIRED <wsa:Action> child element

Implementation of this fault is optional.

6.4.5 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] a QName representing the value S:Receiver

[Subcode] a QName representing the value wsa:EndpointUnavailable

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

[Details] an optional <wsa:RetryAfter> element and an optional <wsa:ProblemIRI> element that conveys the [address] of the [destination].

Implementation of this fault is optional.

7. Security Considerations

Note:

No assumptions are made herein of the application level security requirement, the organization of the application, implementation of senders or receivers, or of the ways that other protocols may make use of WS-Addressing, and what security mechanisms they may employ. A holistic approach to security which considers all components of the application, other protocols utilized, the way that these protocols compose with WS-Security, and the use of other methods or additional techniques is highly recommended.

As discussed in Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*], WS-Addressing supports capabilities that allow a message sender to instruct a message receiver to send additional unsolicited messages to other receivers of their choice and to control the contents of those messages to an extent using reference parameters. The SOAP binding of WS-Addressing transforms EPR reference parameters into SOAP headers and this allows a message sender to request a message receiver to send additional unsolicited SOAP messages to other receivers of their choice and to specify a set of SOAP headers that must be included in such messages.

SOAP headers are a powerful extension mechanism and therefore great care should be taken before honoring a [reply endpoint] or [fault endpoint] to avoid inadvertent participation in the activities of malicious SOAP message senders.

WS-Addressing message addressing properties serialized as SOAP headers (wsa:To, wsa:Action et al.) including those headers present as a result of the [reference parameters] property should be integrity protected as explained in Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*].

Messages that use wsa:ReplyTo or wsa:FaultTo headers whose [address] is not the predefined anonymous URI should include claims that allow a receiver to confirm that the EPR was issued by a principle with authority to represent the [address] of the EPR.

When receiving a SOAP message, certain SOAP headers may have resulted from the serialization of an EPR's [reference parameters] property. A SOAP message receiver should perform additional security and sanity checks to prevent unintended actions.

7.1 Establishing EPR Trust

There are many mechanisms that could be used to supply proof that a message sender has authority to represent the [address] of EPRs supplied within the message. Typically such mechanisms require the inclusion of a WS-Security[*WS-Security [p.22]*] header that contains XML digital signatures binding the wsa:ReplyTo and wsa:FaultTo elements to the SOAP message using a security token issued by an authority trusted by the receiver of the message for the domain of the [address] of the EPR. Possession of a security token issued by a trusted authority for the domain of the [address] of the EPR provides a level of confidence that the message sender has authority to represent the [address].

For example, a message could include a WS-Security[*WS-Security [p.22]*] header that contains XML digital signatures binding the wsa:ReplyTo and wsa:FaultTo elements to the SOAP message using an X.509 certificate for the domain addressed by the [address] of the EPR. If the certificate is issued by a certificate authority trusted by the receiver of the message then the receiver can have some level of confidence that the message sender has authority to represent the [address] of the EPR.

7.2 Additional Security Considerations

The wsa:isReferenceParameter attribute is only meaningful on SOAP headers. Message processors should consider its appearance elsewhere in a SOAP message as a possible attack.

Message processors should consider elements from the soap11, soap12 and wsa namespaces appearing as reference parameters in an EPR as a possible attack.

There are known XML ID and re-structuring attacks which should be considered by message processors, see [*WS-Security [p.22]*] - Security Considerations: Removal and modification of XML elements.

7.3 Additional Considerations for SOAP Intermediaries

To avoid breaking signatures, intermediaries MUST NOT change the XML representation of WS-Addressing headers when relaying those headers. Specifically, intermediaries MUST NOT remove XML content that explicitly indicates otherwise-implied content, and intermediaries MUST NOT insert XML content to make implied values explicit. For instance, if a RelationshipType attribute is present with a value of "http://www.w3.org/2005/08/addressing/reply", an intermediary MUST NOT remove it; similarly, if there is no RelationshipType attribute, an intermediary MUST NOT add one.

8. Conformance

A SOAP 1.2 message conforms to the SOAP 1.2 Addressing 1.0 Module when it contains headers from the wsa namespace, and follows all the constraints on message addressing properties defined by Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*] and by the SOAP 1.2 Addressing 1.0 Module.

A SOAP 1.1 message conforms to the SOAP 1.1 Addressing 1.0 Extension when it contains headers from the wsa namespace, and follows all the constraints on message addressing properties defined by Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*] and by the SOAP 1.1 Addressing 1.0 Extension.

An endpoint which conforms to this specification understands and accepts SOAP messages containing headers in the wsa namespace targeted to it, generates reply or fault messages it may send in response according to the rules outlined in this specification and in Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.21]*].

Note:

Web Services Addressing 1.0 - WSDL Binding[*WS-Addressing-WSDL [p.21]*] defines additional conformance requirements for the description of an endpoint.

Note:

Endpoints MAY accept and respond to messages which contain no WSA headers.

If a receiver processes a message containing a wsa:Action header, this SOAP binding is engaged, and the rules of this specification are in force.

9. References

[WS-Addressing-Core]
Web Services Addressing 1.0 - Core, M. Gudgin, M. Hadley, T. Rogers, Editors. W3C Proposed Recommendation, World Wide Web Consortium, 21 March 2006. This version of the WS-Addressing Core specification is http://www.w3.org/TR/2006/PR-ws-addr-core-20060321. The latest version of WS-Addressing Core is available at http://www.w3.org/TR/ws-addr-core.

[WS-Addressing-WSDL]
Web Services Addressing 1.0 - WSDL Binding, M. Gudgin, M. Hadley, T. Rogers, Ü Yalçinalp, Editors. W3C Working Draft, World Wide Web Consortium, 16 February 2006. This version of the WS-Addressing WSDL Binding specification is http://www.w3.org/TR/2006/WD-ws-addr-wsdl-20060216. The latest version of WS-Addressing WSDL Binding is available at http://www.w3.org/TR/ws-addr-wsdl.

[WSDL 2.0]
Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, R. Chinnici, J. J. Moreau, A. Ryman, S. Weerawarana, Editors. W3C Candidate Recommendation, World Wide Web Consortium, 6 January 2006. This version of the WSDL 2.0 specification is http://www.w3.org/TR/2006/CR-wsdl20-20060106. The latest version of WSDL 2.0 is available at http://www.w3.org/TR/wsdl20.

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

[IETF RFC 3987]
Internationalized Resource Identifiers (IRIs) M. Duerst, M. Suignard. RFC, Internet Engineering Task Force, January 2005. Available at http://www.ietf.org/rfc/rfc3987.txt.

[XML 1.0]
Extensible Markup Language (XML) 1.0 (Third Edition), T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. W3C Recommendation, World Wide Web Consortium, 4 February 2004. This version of the XML 1.0 Recommendation is http://www.w3.org/TR/2004/REC-xml-20040204. 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. W3C Recommendation, 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 (Second Edition), J. Cowan and R. Tobin, Editors. W3C Recommendation, World Wide Web Consortium, 4 February 2004. This version of the XML Information Set Recommendation is http://www.w3.org/TR/2004/REC-xml-infoset-20040204. 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 Second Edition, H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. W3C Recommendation, World Wide Web Consortium, 28 October 2004. This version of the XML Schema Part 1 Recommendation is http://www.w3.org/TR/2004/REC-xmlschema-1-20041028. 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 Second Edition, P. Byron and A. Malhotra, Editors. W3C Recommendation, World Wide Web Consortium, 28 October 2004. This version of the XML Schema Part 2 Recommendation is http://www.w3.org/TR/2004/REC-xmlschema-2-20041028. 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. W3C Recommendation, 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/.

[SOAP 1.2 Part 2: Adjuncts]
SOAP Version 1.2 Part 2: Adjuncts, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. Frystyk Nielsen, Editors. W3C Recommendation, World Wide Web Consortium, 24 June 2003. This version of the "SOAP Version 1.2 Part 2: Adjuncts" Recommendation is http://www.w3.org/TR/2003/REC-soap12-part2-20030624/. The latest version of "SOAP Version 1.2 Part 2: Adjuncts" is available at http://www.w3.org/TR/soap12-part2/.

[SOAP 1.1]
Simple Object Access Protocol (SOAP) 1.1, D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H Frystyk Nielsen, S. Thatte, D. Winer, Editors. W3C Member Submission, 8 May 2000.

[WS-Security]
Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), A. Nadalin, C. Kaler, P. Hallam-Baker, R. Monzillo, Editors. OASIS Standard, OASIS, March 2004.

[SOAP 1.1 Request Optional Response HTTP Binding]
SOAP 1.1 Request Optional Response HTTP Binding, D. Orchard, Editor. World Wide Web Consortium. The latest version of "SOAP 1.1 Request Optional Response HTTP Binding" is available http://www.w3.org/TR/soap11-ror-httpbinding.

A. Acknowledgements (Non-Normative)

This document is the work of the W3C Web Service Addressing Working Group.

Members of the Working Group are (at the time of writing, and by alphabetical order): Abbie Barbir (Nortel Networks), Andreas Bjärlestam (ERICSSON), Dave Chappell (Sonic Software), Eran Chinthaka (WSO2), Francisco Curbera (IBM Corporation), Glen Daniels (Sonic Software), Vikas Deolaliker (Sonoa Systems, Inc.), Paul Downey (BT), Jacques Durand (Fujitsu Limited), Robert Freund (Hitachi, Ltd.), Marc Goodner (Microsoft Corporation), Arun Gupta (Sun Microsystems, Inc.), Hugo Haas (W3C/ERCIM), Marc Hadley (Sun Microsystems, Inc.), David Hull (TIBCO Software, Inc.), Yin-Leng Husband (HP), David Illsley (IBM Corporation), Anish Karmarkar (Oracle Corporation), Paul Knight (Nortel Networks), Philippe Le Hégaret (W3C/MIT), Amelia Lewis (TIBCO Software, Inc.), Bozhong Lin (IONA Technologies, Inc.), Mark Little (JBoss Inc.), Jonathan Marsh (Microsoft Corporation), Jeff Mischkinsky (Oracle Corporation), Nilo Mitra (ERICSSON), Eisaku Nishiyama (Hitachi, Ltd.), Ales Novy (Systinet Inc.), David Orchard (BEA Systems, Inc.), Gilbert Pilz (BEA Systems, Inc.), Tony Rogers (Computer Associates), Tom Rutt (Fujitsu Limited), Davanum Srinivas (WSO2), Jiri Tejkl (Systinet Inc.), Mike Vernal (Microsoft Corporation), Steve Vinoski (IONA Technologies, Inc.), Katy Warr (IBM Corporation), Pete Wenzel (Sun Microsystems, Inc.), Steve Winkler (SAP AG), Ümit Yalçinalp (SAP AG), Prasad Yendluri (webMethods, Inc.).

Previous members of the Working Group were: Lisa Bahler (SAIC - Telcordia Technologies), Rebecca Bergersen (IONA Technologies, Inc.), Ugo Corda (Sun Microsystems, Inc.), Michael Eder (Nokia), Yaron Goland (BEA Systems, Inc.), Marc Goodner (SAP AG), Martin Gudgin (Microsoft Corporation), Mark Nottingham (BEA Systems, Inc.), Mark Peel (Novell, Inc.), Harris Reynolds (webMethods, Inc.), Rich Salz (IBM Corporation), Davanum Srinivas (Computer Associates), Greg Truty (IBM Corporation).

The people who have contributed to discussions on public-ws-addressing@w3.org are also gratefully acknowledged.

B. Change Log (Non-Normative)

B.1 Changes Since Candidate Recommendation

Date	Editor	Description
2006-03-13 @ 22:27	mhadley	Added editorial fixes from Glen
2006-03-13 @ 14:45	mhadley	Added editorial fixes from Hugo
2006-03-10 @ 15:26	mhadley	Added resolution of CR23
2006-03-09 @ 18:36	mhadley	Fixed editorial glitch
2006-03-09 @ 18:35	mhadley	Removed ProblemHeader detail element and made ’Destination Unreachable’, ’Action Not Supported’ and ’Endpoint Unavailable’ faults optional
2006-03-09 @ 18:17	mhadley	Switched Inbound and Outbound to reflect perspective of responder
2006-03-06 @ 19:02	mhadley	Removed redundant ed notes re additional fault codes
2006-03-03 @ 14:10	mhadley	Fixed editor list in references
2006-03-03 @ 09:54	mhadley	Added resolution of CR25 - use property names for SOAP request-response MEP
2006-03-02 @ 14:41	mhadley	Tweaked description of wsa:To with anonymous

2006-03-02 @ 10:14	mhadley	Added definition of term ’response endpoint’
2006-03-02 @ 08:51	mhadley	Added resolution to CR24, finessing text about use of predefined fault actions
2006-03-01 @ 17:01	mhadley	Added CR20 resolution - Strengthen guidance on protocol-specifc fault action values
2006-03-01 @ 16:52	mhadley	Added CR18 resolution. Also fixed some terminology ’response endpoint’ to ’[reply endpoint] property, changed explicit URIs to use predefined entity and fixed use of invalid markup ’<a>’ to ’<xspecref>’
2006-02-18 @ 23:38	trogers	Added editorial note in place of a reference to the WG Note which will define the SOAP 1.1/HTTP request/optional-response MEP.
2006-02-08 @ 07:05	trogers	Changed Change Log limit from 20060101 to 20061201
2006-02-08 @ 06:48	trogers	Added the resolution of CR15/i067/i068 - discussion of anonymous/non-anonymous in SOAP 1.1/SOAP 1.2.
2006-02-08 @ 06:23	trogers	Added the resolution of CR14 - stating independence of SOAP headers and transport properties
2006-02-08 @ 06:13	trogers	Added the resolution of CR13 - adding the OnlyAnonymousSupported and OnlyNonAnonymousSupported subsubcodes
2005-11-22 @ 20:41	mhadley	Added issue cr11 resolution, minor editorial teaks to SOAPAction to [action] relationship text
2005-11-08 @ 05:21	trogers	Implemented the resolution of CR7, stating when WSA is engaged.
2005-11-07 @ 07:18	mhadley	Added resolution to issue cr9, added URI for use with generic SOAP faults
2005-11-06 @ 12:48	trogers	Incorporated the resolution of CR8 by rephrasing the requirements on the SOAPAction HTTP header for SOAP 1.1.
2005-11-06 @ 12:30	trogers	Incorporated the resolution of CR4 by extending the definition of the anonymous URI.
2005-10-24 @ 05:02	trogers	Removed ProblemIRI from InvalidAddress explanation, resolving Issue CR6
2005-10-17 @ 18:44	mhadley	Added namesapce change policy
2005-09-15 @ 19:42	mhadley	Added new section for post CR changes

2005-09-15 @ 18:27	mhadley	Added resolution to issue cr1 - changed ActionMismatch to ProblemAction

B.2 Changes Since Last Call Working Draft

Date	Editor	Description
2005-07-20 @ 19:04	mhadley	Added ednote asking for feedback on removal of [source endpoint] and wsa:From
2005-07-20 @ 18:21	mhadley	Added resolution to issues lc55 and lc87 - reworked security section
2005-07-20 @ 15:53	mhadley	Added resolution to issue lc76 - expanded faults section
2005-07-19 @ 20:08	mhadley	Added partial resolution to issue lc76 - added new sections for standard detail items and grouped faults in new section
2005-07-19 @ 18:46	mhadley	Added revised resolution to issue lc20 - clarified meaning of anonymous uri in SOAP
2005-06-21 @ 17:12	mhadley	Added issue 71 resolution - clarified that the value of reason text is recommended but not required
2005-06-14 @ 14:25	mhadley	Added resolutions to issues lc56 and lc72 - Added new fault detail elements and header block for SOAP 1.1
2005-06-03 @ 20:36	mhadley	Fixed typo in document prologue
2005-06-03 @ 20:33	mhadley	Added resolutions to issues lc58, lc79, lc91, lc102
2005-06-02 @ 19:45	mhadley	Added resolution to issue lc62 - added note confirming that endpoints may consume and respond to messages that do not use any WS-Addr headers
2005-06-02 @ 19:12	mhadley	Added resolution to issue lc6 and lc35 - added new conformance section, moved conformance text from module and extension sections
2005-06-02 @ 18:56	mhadley	Added resolution to issue lc73 - added note warning about use of reference parameters conflicting with normal message semantics
2005-06-02 @ 18:15	mhadley	Added resolution to issue lc37 - added DOS attack security considerations
2005-06-02 @ 17:43	mhadley	Added clarifications of fault property values

2005-05-25 @ 21:40	mhadley	Added new section in changelog to account for previous draft publication
2005-05-25 @ 21:20	mhadley	Added resolution to issue lc105 - added requirement that no additional %-escaping be peformed on IRI type message addressing properties when serialized
2005-05-25 @ 21:07	mhadley	Added resolution to issue lc73 - clarrified meaning of omitting RetryAfter
2005-05-25 @ 21:03	mhadley	Added resolution to issue lc57 - added normative text describing fault binding
2005-05-25 @ 20:20	mhadley	Added resolution to issue lc66 - made it clear that type often refers to the content of elements rather than the element as a whole which can often also include attributes
2005-05-18 @ 19:44	mhadley	Added lc59 resolution - added missing namespace declaration in example
2005-05-18 @ 19:42	mhadley	Added lc53 resolution - expanded MAP to message addressing property and fixed editorial glitch
2005-05-18 @ 19:37	mhadley	Added lc52 resolution - MessageId to MessageID
2005-05-18 @ 19:35	mhadley	Added lc51 resolution - reordered property list to match order in core
2005-05-18 @ 19:22	mhadley	Added lc47 resolution - fixed URL in WSDL 2.0 biblio entry
2005-05-18 @ 19:16	mhadley	Added lc38 resolution - nonNegativeInteger to unsignedLong for RetryAfter
2005-05-18 @ 18:03	mhadley	Added lc67 resolution - made namespace uri a link
2005-05-18 @ 17:58	mhadley	Added lc64 resolution - numerous editorial fixes
2005-05-16 @ 20:20	mgudgin	Fixed reference to RFC3987 to match format of other biblio entries
2005-05-13 @ 18:56	mhadley	Added resolutions to issues 33 and 34: editorial corrections to binding MAP to SOAP headers and new rule against multiple headers targetted at same recipient
2005-05-05 @ 18:10	mhadley	Added issue 28 resolution: fixed use of mixed notation and indirect terminology for MAPs in Binding Message Addressing Properties section

2005-05-05 @ 17:39	mhadley	Added resolution to issues 26 and 36: Clarified use of invalid map fault for mismatched wsa:Action and SOAPAction; renamed and clarified invalid map and missing map faults.
2005-04-22 @ 20:01	mhadley	Added resolution to lc32 - added note warning of infoset changes due to IsReferenceParameter addition when binding [reference parameter] to SOAP.
2005-04-22 @ 19:51	mhadley	Added resolution to lc31 - clarified what to do if a reference parameter already has an IsReferenceParameter attribute.
2005-04-22 @ 19:46	mhadley	Added resolution to lc30 - added new section for definition of IsReferenceParameter attribute.
2005-04-22 @ 19:26	mhadley	Added resolution to lc29 - capitalized first character of IsReferenceParameter attribute.
2005-04-22 @ 19:07	mhadley	Added resolution to lc27 - clarified confusing use of XML infoset terminology in XML representation of properties.
2005-04-22 @ 18:58	mhadley	Added resolution to lc24 - editorial nits.
2005-04-22 @ 18:49	mhadley	Added resolution to lc23 - changed IRI to URI for constant values that are URIs.
2005-04-22 @ 15:27	mhadley	Added resolution to lc1 - clarified impact of omitting [message id], [reply endpoint] and [fault endpoint] on fault message generation
2005-04-12 @ 13:17	mhadley	Fixed closing element in example

B.3 Changes Since Second Working Draft

Date	Editor	Description
2005-03-21 @ 23:15	mgudgin	Added sentence about SOAP 1.1 to section 4
2005-03-18 @ 23:21	mgudgin	s/Addresssing/Addressing
2005-03-10 @ 03:40	mhadley	Incorporated additional editorial fixes from J. Marsh.
2005-03-10 @ 03:16	mhadley	Incorporated additional issue resolution text for issues 7 and 44 from H. Haas.
2005-03-10 @ 02:06	mhadley	Incorporated editorial fixes from J. Marsh.
2005-03-09 @ 07:11	mhadley	Fixed example that didn't reflect the chnage from wsa:Type to wsa:isReferenceParameter
2005-03-08 @ 20:50	mhadley	Added resolution to issue 53 (schema tweaks)
2005-03-02 @ 21:18	mhadley	Added resolution to issue 4
2005-03-02 @ 20:30	mhadley	Added resolution to issue 7
2005-03-02 @ 19:36	mhadley	Added resolution to issues 22 and 51/
2005-02-28 @ 22:08	mhadley	Added resolution to issues 24 and 26
2005-02-27 @ 19:42	mhadley	Changed URI to IRI where appropriate.
2005-02-17 @ 15:37	mhadley	Added issue 47 resolution
2005-02-15 @ 22:06	mhadley	Fixed some references to message information headers to message information properties

B.4 Changes Since First Working Draft

Date	Editor	Description
2005-02-01 @ 19:49	mhadley	Removed several occurances of the word ’identify’ when used with endpoint references. Replaced with ’reference’ or ’address’ as appropriate.
2005-01-24 @ 20:22	mgudgin	Removed spurious reference to section 3.3.2 from Section 3
2005-01-23 @ 21:11	mgudgin	Incorporated resolution of issue i008; added wsa:Type attribute to reference parameters
2005-01-20 @ 13:10	mgudgin	Removed text from first paragraph of section 3 per resolution of issue i040
2005-01-16 @ 22:41	mgudgin	s/PortType/InterfaceName in certain examples
2004-12-16 @ 18:20	mhadley	Added resolution to issue 19 - WSDL version neutrality
2004-12-16 @ 16:50	mhadley	Added issue 33 resolution
2004-12-14 @ 20:10	mhadley	Switched back to edcopy formatting
2004-12-14 @ 20:02	mhadley	Enhanced auto-changelog generation to allow specification of data ranges for logs. Split change log to show changes between early draft and first working draft and changes since first working draft.
2004-12-14 @ 18:13	mhadley	Added resolutions for issues 12 (EPR lifecycle), 37 (relationship from QName to URI) and 39 (spec name versioning)

B.5 Changes Since Submission

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