Web Services Addressing 1.0 - WSDL Binding

W3C Working Draft 13 April 2005

This version:
http://www.w3.org/TR/2005/WD-ws-addr-wsdl-20050413
Latest version:
http://www.w3.org/TR/ws-addr-wsdl
Previous versions:
http://www.w3.org/TR/2005/WD-ws-addr-wsdl-20050215

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 1.0 - WSDL Binding (this document) defines how the abstract properties defined in Web Services Addressing 1.0 - 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 third Public Working Draft of the Web Services Addressing 1.0 - 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.

This Working Draft reflects the current position of the Working Group. This version of the draft shows, among other things, how to use the [metadata] message addressing property of Web Services Addressing 1.0 to provide WSDL information. 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.20] . A list of remaining issues is also available.

Discussion of this document takes place on the 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.

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.	

>

Table of Contents<a>

Appendices

>

>1. Introduction

Web Services Addressing 1.0 - Core[*WS-Addressing-Core [p.18]*] defines a set of abstract properties and an XML Infoset [*XML Information Set [p.19]*] representation thereof to reference Web service endpoints and to facilitate end-to-end addressing of endpoints in messages. Web Services Addressing 1.0 - WSDL Binding (this document) defines how the abstract properties defined in Web Services Addressing 1.0 - Core are described using WSDL. WS-Addressing is designed to be able to work with WSDL 2.0 [*WSDL 2.0 [p.18]*] and also (for backwards compatibility with WSDL 1.1 [*WSDL 1.1 [p.19]*] described services.

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.18]*].

When describing abstract data models, this specification uses the notational convention used by the XML Infoset [*XML Information Set [p.19]*]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [*XML Schema Structures [p.19]* , *XML Schema Datatypes [p.19]*], this specification uses the notational convention of WS-Security [*WS-Security [p.19]*]. 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.19]*]).

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/2005/03/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.19]*]. WS-Addressing may be used with SOAP [*SOAP 1.2 Part 1: Messaging Framework [p.19]*] as described in Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing-SOAP [p.18]*]. The examples in this specification use an XML 1.0 [*XML 1.0 [p.18]*] representation but this is not a requirement.

All information items defined by this specification are identified by the XML namespace URI [*XML Namespaces [p.19]*] "http://www.w3.org/2005/03/addressing/wsdl". A normative XML Schema [*XML Schema Structures [p.19]* , *XML Schema Datatypes [p.19]*] document can be obtained by dereferencing the XML namespace URI.

2. Including WSDL Metadata in EPRs

An EPRs metadata section can contain a reference to WSDL metadata or can include embedded WSDL metadata.

2.1 Referencing WSDL Metadata from an EPR

The WSDL binding of Web Services Addressing introduces the following element and attribute information items for referencing WSDL metadata from an EPRs metadata section:

wsaw:InterfaceName

> A QName identifying a description of the sequences of messages that a service sends and/or receives. This corresponds to a WSDL 2.0 interface and/or a WSDL 1.1 port type.

wsaw:ServiceName

A QName that identifies the set of endpoints at which a particular Web service is deployed. The set of endpoints is represented by a service in WSDL 2.0 and WSDL 1.1.

wsaw:ServiceName/@EndpointName

An NCName that identifies one endpoint amongst the set identified by by the service name above. An endpoint is represented by an endpoint in WSDL 2.0 and a port in WSDL 1.1.

The element information items defined above are used in an EPRs metadata section. The following shows an example endpoint reference. This references the interface named "fabrikam:Inventory" at the endpoint IRI "http://example.com/www.fabrikam/acct". Note the use of the WSDL[*WSDL 2.0 [p.18]*] wsdlLocation attribute.

Example 2-1. Example endpoint reference.

```
<wsa:EndpointReference
    xmlns:wsa="http://www.w3.org/2005/03/addressing"
    xmlns:fabrikam="http://example.com/fabrikam">
  <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
  <wsa:Metadata
      xmlns:wsdli="http://www.w3.org/2004/08/wsdl-instance"
      wsdli:wsdlLocation="http://example.com/fabrikam http://example.com/fabrikam.wsdl">
    <wsaw:InterfaceName>fabrikam:Inventory</wsaw:InterfaceName>
  </wsa:Metadata>
</wsa:EndpointReference>
```

2.2 Embedding WSDL Metadata in an EPR

WSDL 1.1 or 2.0 definitions can be embedded in the metadata section of an EPR to provide a consuming application with WSDL information that applies to the referenced endpoint. To do so, the creator of an EPR MAY include a WSDL 2.0 description element (or a WSDL 1.1 definitions element) in the metadata property of the EPR. The semantics of the embedded WSDL is as defined by the WSDL 2.0 or 1.1 specifications.

In particular, embedding a WSDL service component description MAY be used by EPR issuers to indicate the presence of alternative addresses and protocol bindings to access the referenced endpoint. The alternatives are provided by the different endpoints of the embedded service. In the case of WSDL 1.1, additional ports may be conveyed by the WSDL 1.1 service definition which are not alternative access channels to the endpoint. In that case, if the InterfaceName or ServiceName elements are also included in the metadata section of the EPR, only the ports with the same interface as that specified are to be considered alternative access channels.

If the ServiceName element appears in the EPR’s [metadata] and an embedded WSDL service component is also provided inside a descriptions or definitions component, then the ServiceName SHOULD match the name of (one or more of) the WSDL service(s) included therein; the endpoint (port) name SHOULD match as well if present. The behavior of an EPR consumer when the ServiceName doesn’t match an embedded description is undefined.

Example 2-2. An EPR containing WSDL 2.0 metadata

```
<wsa:EndpointReference
    xmlns:wsa="http://www.w3.org/2005/03/addressing">

  <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
  <wsa:Metadata
      xmlns:wsdli="http://www.w3.org/2004/08/wsdl-instance"
      wsdli:wsdlLocation="http://example.com/fabrikam http://example.com/fabrikam.wsdl">
    <wsdl20:description
        targetNamespace="http://example.com/fabrikam"
        xmlns:fabrikam="http://example.com/fabrikam"
        xmlns:abc="http://www.abccorp.com/"
        xmlns:wsdl20="http://www.w3.org/2004/08/wsdl">
      <wsdl20:import namespace="http://example.com/fabrikam/"
          location="http://example.com/fabrikam/fabrikam.wsdl/"/>
      <wsdl20:import namespace="http://www.abccorp.com/"
          location="http://www.abccorp.com/abc.wsdl"/>
      <wsdl20:service name="InventoryService"
          interface="fabrikam:Inventory">
        <wsdl20:endpoint name="ep1" binding="abc:soap-http-binding"
            address="http://example.com/fabrikam/acct"/>
        <wsdl20:endpoint name="ep2" binding="abc:iiop"
            address="..."/>
        <wsdl20:endpoint name="ep3"
            binding="abc:soap-http-binding"
            address="http://alt.example.com/fabrikam/acct"/>
      </wsdl20:service>
    </wsdl20:description>
  </wsa:Metadata>
</wsa:EndpointReference>
```

Example 2-3. An EPR containing WSDL 1.1 metadata

```
<wsa:EndpointReference
    xmlns:wsa="http://www.w3.org/2005/03/addressing">
  <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
  <wsa:Metadata>
    <wsdl11:definitions targetNamespace="http://example.com/fabrikam"
        xmlns:fabrikam="http://example.com/fabrikam"
        xmlns:abc="http://www.abccorp.com/"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
        xmlns:iiop="http://www.iiop.org/"
        xmlns:wsdl11="http://schemas.xmlsoap.org/wsdl/">
      <wsdl11:import namespace="http://example.com/fabrikam"
          location="http://example.com/fabrikam/fabrikam.wsdl"/>
      <wsdl11:import namespace="http://www.abccorp.com/"
          location="http://www.abccorp.com/abc.wsdl"/>
      <wsdl11:service name="InventoryService">
        <wsdl11:port name="ep1" binding="abc:soap-http-binding">
          <soap:address location="http://example.com/fabrikam/acct"/>
        </wsdl11:port>
        <wsdl11:port name="ep2" binding="abc:iiop">
          <iiop:address location="..."/>
        </wsdl11:port>
      </wsdl11:service>
    </wsdl11:definitions>
  </wsa:Metadata>
</wsd:EndpointReference>
```

3. Associating Action with WSDL Operations

Ensuring that there is sufficient information within a message to distinguish which WSDL operation it is associated with is specified as a best practice in WSDL 2.0*WSDL 2.0 [p.18]* . The [action] property provides a mechanism to fulfill that best practice.

WS-Addressing defines two mechanisms to associate a value of the [action] property with input, output and fault elements within a WSDL description: explicit and defaulting. Explicit association is described in section **3.1 Explicit Association** [p.7] ; action defaulting (where a unique value for the [action] property is automatically generated) is described in section **3.3 Default Action Pattern for WSDL 1.1** [p.10] for WSDL 1.1 and section **3.2 Default Action Pattern for WSDL 2.0** [p.8] for WSDL 2.0.

3.1 Explicit Association

WS-Addressing defines a global attribute, wsa:Action, that may be used to explicitly define the value of the [action] property for messages in a WSDL description. The type of the attribute is xs:anyURI and it is used as an extension on the WSDL input, output and fault elements.

For example consider the following WSDL excerpt:

Example 3-1. Explicit specification of wsa:Action value in a WSDL 2.0 description.

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <interface name="StockQuoteInterface">
    <operation name="GetLastTradePrice" pattern="http://www.w3.org/2004/08/wsdl/in-out">
      <input element="tns:GetTradePricesInput" messageLabel="In"
            wsa:Action="http://example.com/GetQuote"/>
      <output element="tns:GetTradePricesOutput" messageLabel="Out"
            wsa:Action="http://example.com/Quote"/>
    </operation>
  </interface>
  ...
</definitions>
```

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

Example 3-2. Explicit specification of wsa:Action value in a WSDL 1.1 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 for WSDL 2.0

In the absence of the wsa:Action attribute, the following pattern is used in WSDL 2.0 documents to construct a default action for inputs and outputs. The general form of an action URI is as follows:

Example 3-3. Structure of defaulted wsa:Action IRI in WSDL 2.0.

```
[target namespace][delimiter][interface name][delimiter][operation name][direction token]
```

For fault messages, the general form of an action IRI is as follows:

Example 3-4. Structure of default wsa:Action IRI for faults

```
[target namespace][delimiter][interface name][delimiter][fault name]
```

Where:

[delimiter]

> is ":" when the [target namespace] is a URN, otherwise "/". Note that for IRI schemes other than URNs which aren’t path-based (i.e. those that outlaw the "/" character), the default action value may not conform to the rules of the IRI scheme. Authors are advised to specify explicit values in the WSDL in this case.

[target namespace]

> is the {target namespace} of the interface. If [target namespace] ends with a "/" an additional "/" is not added.

[interface name]

> is the {name} of the interface.

[operation name]

> is the {name} of the operation.

[fault name]

is the {name} of the fault.

[direction token]

- Empty ("") where the operation’s {message exchange pattern} is "http://www.w3.org/2004/08/wsdl/in-only", "http://www.w3.org/2004/08/wsdl/robust-in-only", "http://www.w3.org/2004/08/wsdl/out-only", or "http://www.w3.org/2004/08/wsdl/robust-out-only".

- "Request" where the operation’s {message exchange pattern} is "http://www.w3.org/2004/08/wsdl/in-out" or "http://www.w3.org/2004/08/wsdl/in-opt-out" and the message reference’s {message label} = ’in’.

- "Solicit" where the operation’s {message exchange pattern} is "http://www.w3.org/2004/08/wsdl/out-in" or "http://www.w3.org/2004/08/wsdl/out-opt-in" and the message reference’s {message label} = ’out’.

- "Response" where the operation’s {message exchange pattern} is "http://www.w3.org/2004/08/wsdl/in-out" or "http://www.w3.org/2004/08/wsdl/in-opt-out" and the message reference’s {message label} = ’out’.

- "Response" where the operation’s {message exchange pattern} is "http://www.w3.org/2004/08/wsdl/out-in", or "http://www.w3.org/2004/08/wsdl/out-opt-in" and the message reference’s {message label} = ’in’.

- {message label} where the {message exchange pattern} is not one of the MEP IRIs defined in WSDL 2.0 Part 2.

For example consider the following WSDL excerpt:

Example 3-5. Example WSDL without explicit wsa:Action values with explicit message names.

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <interface name="StockQuoteInterface">
    <operation name="GetLastTradePrice" pattern="http://www.w3.org/2004/08/wsdl/in-out">
      <input element="tns:GetTradePricesInput" messageLabel="in" name="GetQuote"/>
      <output element="tns:GetTradePricesOutput" messageLabel="out" name="Quote"/>
    </operation>
  </portType>
  ...
</definitions>
```

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

[interface name] = StockQuoteInterface

[operation name] = GetLastTradePrice

[direction token] for input is Request

[direction token] for output is Response

Applying the pattern above with these values we have:

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

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

3.3 Default Action Pattern for WSDL 1.1

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 IRI is as follows:

Example 3-6. Structure of defaulted wsa:Action IRI.

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

For fault messages, the general form of an action IRI is as follows:

Example 3-7. Structure of default wsa:Action IRI for faults

```
[target namespace][delimiter][port type name][delimiter][operation name]Fault:[fault name]
```

Where:

[delimiter]

> is ":" when the [target namespace] is a URN, otherwise "/". Note that for IRI schemes other than URNs which aren’t path-based (i.e. those that outlaw the "/" character), the default action value may not conform to the rules of the IRI scheme. Authors are advised to specify explicit values in the WSDL in this case.

"Fault:"

> is a literal character string to be included in the action.

[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.

[fault name]

is the name of the fault (/definition/porttype/operation/fault/@name).

For example consider the following WSDL excerpt:

Example 3-8. 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"/>
          <fault message="tns:ErrorMessage" name="Error"/>
    </operation>
  </portType>
  ...
</definitions>
```

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

[port type name] = StockQuotePortType

[input name] = GetQuote

[output name] = Quote

[fault name] = Error

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

fault action = http://example.com/stockquote/StockQuotePortType/GetLastTradePriceFault:Error

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

Example 3-9. 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 Section 2.4.5 of WSDL 1.1, 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

This section describes which of the core message properties are mandatory or optional for messages in the various MEPs defined by WSDL 1.1 and WSDL 2.0.

4.1 WSDL 1.1 Message Exchange Patterns

This section describes which of the core message properties are mandatory or optional for messages in the various MEPs defined by WSDL 1.1.

4.1.1 One-way

This is a straightforward one-way message. No responses are expected but related messages could be sent as part of other message exchanges.

Table 4-1. Message addressing properties for one way message.

Property	Mandatory	Description
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	N	Intended receiver for replies to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[fault endpoint]	N	Intended receiver for faults related to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[message id]	N	Unique identifier for this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[relationship]	N	Indicates relationship to a prior message. Unused in this MEP, but may be included to facilitate longer running message exchanges.

4.1.2 Request-Response

This is request-response. A reply is expected hence mandating [reply endpoint] in the request message. The response message might be a fault.

Table 4-2. Message addressing properties for request message.

Property	Mandatory	Description
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	Y	Intended receiver for the reply to this message.
[fault endpoint]	N	Intended receiver for faults related to this message. May be included to direct fault messages to a different endpoint than [reply endpoint].
[message id]	Y	Unique identifier for this message. Used in the [relationship] property of the reply message.
[relationship]	N	Indicates relationship to a prior message. Unused in this MEP, but may be included to facilitate longer running message exchanges.

Table 4-3. Message addressing properties for response message.

Property	Mandatory	Description
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	N	Intended receiver for replies to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[fault endpoint]	N	Intended receiver for faults related to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[message id]	N	Unique identifier for this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[relationship]	Y	Indicates that this message is a reply to the request message using the request message [message id] value and the predefined `http://www.w3.org/2005/03/addressing/reply` IRI.

4.1.3 Notification

From the WS-Addressing perspective this MEP is the same as One-way. The properties defined in **4.1.1 One-way** [p.12] apply to this MEP also.

4.1.4 Solicit-response

From the WS-Addressing perspective this MEP is the same as Request-response. The properties defined in **4.1.2 Request-Response** [p.13] apply to this MEP also.

4.2 WSDL 2.0 Message Exchange Patterns

This section describes which of the core message properties are mandatory or optional for messages in the various MEPs defined by WSDL 2.0.

4.2.1 In-only

This is a straightforward one-way message. No responses are expected but related messages could be sent as part of other message exchanges.

Table 4-4. Message addressing properties for in message.

Property	**Mandatory**	**Description**
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	N	Intended receiver for replies to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[fault endpoint]	N	Intended receiver for faults related to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[message id]	N	Unique identifier for this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[relationship]	N	Indicates relationship to a prior message. Unused in this MEP, but may be included to facilitate longer running message exchanges.

4.2.2 Robust In-only

This one-way MEP allows fault messages. The [message id] property is needed in the initial message in order to be able to correlate any fault with that message.

Table 4-5. Message addressing properties for in message.

Property	**Mandatory**	**Description**
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	N	Intended receiver for replies to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[fault endpoint]	N	Intended receiver for faults related to this message.
[message id]	Y	Unique identifier for this message. Used in the [relationship] property of any resulting fault message.
[relationship]	N	Indicates relationship to a prior message. Unused in this MEP, but may be included to facilitate longer running message exchanges.

Editorial note: MJH	
Given that [message id] is mandatory above, should [fault endpoint] also be required ?	

Table 4-6. Message addressing properties for fault message.

Property	**Mandatory**	**Description**
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	N	Intended receiver for replies to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[fault endpoint]	N	Intended receiver for faults related to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[message id]	N	Unique identifier for this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[relationship]	Y	Indicates that this message is a response to the in message using the in message [message id] value and the predefined `http://www.w3.org/2005/03/addressing/reply` IRI.

Editorial note: MJH	
Should [fault endpoint] be prohibited in the above ? This would prevent getting a fault in response to a fault.	

4.2.3 In-out

This is a two-way MEP. A reply is expected hence mandating [reply endpoint] in the request message. The response message might be a fault.

Table 4-7. Message addressing properties for in message.

Property	**Mandatory**	**Description**
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	Y	Intended receiver for the reply to this message.
[fault endpoint]	N	Intended receiver for faults related to this message. May be included to direct fault messages to a different endpoint than [reply endpoint].
[message id]	Y	Unique identifier for this message. Used in the [relationship] property of the out message.
[relationship]	N	Indicates relationship to a prior message. Unused in this MEP, but may be included to facilitate longer running message exchanges.

Table 4-8. Message addressing properties for out message.

Property	**Mandatory**	**Description**
[destination]	Y	Provides the address of the intended receiver of this message
[action]	Y	Identifies the semantics implied by this message
[source endpoint]	N	Message origin. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[reply endpoint]	N	Intended receiver for replies to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[fault endpoint]	N	Intended receiver for faults related to this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[message id]	N	Unique identifier for this message. Unused in this MEP, but may be included to facilitate longer running message exchanges.
[relationship]	Y	Indicates that this message is a response to the in message using the in message [message id] value and the predefined `http://www.w3.org/2005/03/addressing/reply` IRI.

4.2.4 In-optional-out

This MEP differs from the In-out MEP in that the subsequent message is optional. This difference doesn’t affect the message properties so the properties defined in **4.2.3 In-out** [p.16] apply to this MEP also

4.2.5 Out-only

From the WS-Addressing perspective this MEP is the same as In-only. The properties defined in **4.2.1 In-only** [p.14] apply to this MEP also.

4.2.6 Robust Out-only

From the WS-Addressing perspective this MEP is the same as Robust In-only. The properties defined in **4.2.2 Robust In-only** [p.15] apply to this MEP also.

4.2.7 Out-in

From the WS-Addressing perspective this MEP is the same as In-out. The properties defined in **4.2.3 In-out** [p.16] apply to this MEP also.

4.2.8 Out-optional-in

This MEP differs from the Out-in MEP in that the subsequent message is optional. This difference doesn’t affect the message properties so the properties defined in **4.2.3 In-out** [p.16] apply to this MEP also

5. References

[WS-Addressing-Core]
Web Services Addressing 1.0 - Core, M. Gudgin, M. Hadley, Editors.

[WS-Addressing-SOAP]
Web Services Addressing 1.0 - SOAP Binding, M. Gudgin, M. Hadley, Editors.

[WSDL 2.0]
Web Services Description Language 2.0, R. Chinnici, M. Gudgin, J. J. Moreau, J. Schlimmer, S. Weerawarana, Editors. World Wide Web Consortium, 3 August 2004. This version of the WSDL 2.0 specification is http://www.w3.org/TR/2004/WD-wsdl20-20040803. 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. Internet Engineering Task Force, June 1999. Available at http://www.ietf.org/rfc/rfc2119.txt.

[RFC 3987]
M. Duerst, M. Suignard, "Internationalized Resource Identifiers (IRIs)", January 2005. (See *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. 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. 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. 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. 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. 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. 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)

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), Rebecca Bergersen (IONA Technologies, Inc.), Andreas Bjärlestam (ERICSSON), Ugo Corda (SeeBeyond Technology Corporation), Francisco Curbera (IBM Corporation), Glen Daniels (Sonic Software), Paul Downey (BT), Jacques Durand (Fujitsu Limited), Michael Eder (Nokia), Robert Freund (Hitachi, Ltd.), Yaron Goland (BEA Systems, Inc.), Martin Gudgin (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), Anish Karmarkar (Oracle Corporation), Paul Knight (Nortel Networks), Philippe Le Hégaret (W3C/MIT), Mark Little (Arjuna Technologies Ltd.), Jonathan Marsh (Microsoft Corporation), Jeff Mischkinsky (Oracle Corporation), Nilo Mitra (ERICSSON), Eisaku Nishiyama (Hitachi, Ltd.), Mark Nottingham (BEA Systems, Inc.), Ales Novy (Systinet Inc.), David Orchard (BEA Systems, Inc.), Mark Peel (Novell, Inc.), Tony Rogers (Computer

Associates), Tom Rutt (Fujitsu Limited), Rich Salz (DataPower Technology, Inc.), Davanum Srinivas (Computer Associates), Jiri Tejkl (Systinet Inc.), Greg Truty (IBM Corporation), Steve Vinoski (IONA Technologies, Inc.), Pete Wenzel (SeeBeyond Technology Corporation), Steve Winkler (SAP AG), Ümit Yalç&#305;nalp (SAP AG), Prasad Yendluri (webMethods, Inc.).

Previous members of the Working Group were: Lisa Bahler (SAIC - Telcordia Technologies), Marc Goodner (SAP AG), Harris Reynolds (webMethods, Inc.).

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 Second Working Draft

Date	Editor	Description
2005-03-21 @ 23:15	mgudgin	Moved sentence on WSDL 2.0/WSDL 1.1 from Section 1.2 to Section 1
2005-03-10 @ 03:40	mhadley	Incorporated additional editorial fixes from J. Marsh.
2005-03-10 @ 02:06	mhadley	Incorporated editorial fixes from J. Marsh.
2005-03-02 @ 21:22	mhadley	Fixed some problems with use of wsdli:wsdlLocation.
2005-03-01 @ 13:33	mhadley	Changed MUST to SHOULD in section 2.2 wrt matching port name
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.
2005-02-23 @ 16:11	mhadley	Incorporated resolution to issue 17b
2005-02-15 @ 23:19	mhadley	Added resolution to issue 45

B.2 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.
2005-01-25 @ 22:23	mhadley	Added descriptive text for wsa:Action attribute. Fixed references to WSDL 1.1 to be more explicit version-wise.
2005-01-24 @ 10:12	mgudgin	Incorporated resolution of i034 and i035; default action URI for WSDL 2.0 and default action URI for faults. All edits in section 3
2005-01-18 @ 04:01	mgudgin	Modified text in Section 2 WRT closing issue i020
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.3 Changes Since Submission

Date	Editor	Description
2004-12-04 @ 02:04	mgudgin	Added text to section on WSDL MEPs per resolution of Issue i003
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