Web Services Addressing 1.0 - Core

W3C Recommendation 9 May 2006

This version:
http://www.w3.org/TR/2006/REC-ws-addr-core-20060509
Latest version:
http://www.w3.org/TR/ws-addr-core
Previous versions:
http://www.w3.org/TR/2006/PR-ws-addr-core-20060321
Editors:
Martin Gudgin, Microsoft Corp
Marc Hadley, Sun Microsystems, Inc
Tony Rogers, Computer Associates International, Inc

Please refer to the **errata** for this document, which may include some normative corrections.

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

See also **translations**.

Abstract

Web Services Addressing provides transport-neutral mechanisms to address Web services and messages. Web Services Addressing 1.0 - Core (this document) defines a set of abstract properties and an XML Infoset [*XML Information Set [p.17]*] representation thereof to reference Web services and to facilitate end-to-end addressing of endpoints in messages. This specification enables messaging systems to support message transmission through networks that include processing nodes such as endpoint managers, firewalls, and gateways in a transport-neutral manner.

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 Recommendation of the Web Services Addressing 1.0 - Core specification. It has been produced by the Web Services Addressing Working Group (WG), which is part of the W3C Web Services Activity.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

The Working Group made the following editorial changes to the Proposed Recommendation in response to comments: normative and informative references are now more clearly distinguished, and some typographical errors were fixed. 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.

Please report errors in this document to the public public-ws-addressing-comments@w3.org mailing list (public archive).

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.

Table of Contents

Appendix

---

1. Introduction

Web Services Addressing 1.0 - Core (WS-Addressing) defines two constructs, message addressing properties and endpoint references, that normalize the information typically provided by transport protocols and messaging systems in a way that is independent of any particular transport or messaging system.

A Web service endpoint is a (referenceable) entity, processor, or resource to which Web service messages can be addressed. Endpoint references convey the information needed to address a Web service endpoint. Note that WSDL 2.0 has an Endpoint component [*WSDL 2.0 Core Language [p.18]* , Section 2.15 Endpoint] which along with other WSDL 2.0 components can be used to describe a Web service endpoint. A Web service endpoint may in fact have multiple such descriptions. Similarly, multiple endpoint references (EPRs) can be used to convey information needed to address a particular Web service endpoint. An EPR is intended to convey information required to address a Web service endpoint whereas a WSDL 2.0 description is intended to describe a Web service.

This specification defines a family of message addressing properties that convey end-to-end message characteristics including references for source and destination endpoints and message identity that allows uniform addressing of messages independent of the underlying transport.

Both of these constructs are designed to be extensible and re-usable so that other specifications can build on and leverage endpoint references and message addressing properties.

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 (see Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing SOAP Binding [p.18]*] for more information on the use of WS-Addressing in SOAP):

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 header blocks. Specifically, line (02) 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.17]*].

When describing abstract data models, this specification uses the notational convention used by the XML Infoset [*XML Information Set [p.17]*]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [*XML Schema Structures [p.17]* , *XML Schema Datatypes [p.17]*], this specification uses the notational convention of WS-Security [*WS-Security 2004 [p.18]*]. Specifically, each member of an Element Information Item'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/>).

Where pseudo-schemas are provided for a component, they use BNF-style conventions for attributes and elements: "?" denotes optionality (i.e. zero or one occurrences), "*" denotes zero or more occurrences, "+" one or more occurrences, "[" and "]" are used to form groups, and "|" represents choice. Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema. Elements with simple content are conventionally assigned a value which corresponds to the type of their content, as defined in the normative schema. Pseudo schemas do not include extensibility points for brevity.

When defining the cardinality of endpoint reference properties and message addressing properties, this specification uses the following notation: (*n..m*), where *n* is the minimum allowed number of occurrences of the property and *m* is the maximum allowed number of occurrences. When *n* has the same value as *m* then exactly that number of occurrences of the property must be present in the associated endpoint reference or message.

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.17]*]).

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

Prefix	Namespace
wsa	http://www.w3.org/2005/08/addressing
S	http://www.w3.org/2003/05/soap-envelope
xs	http://www.w3.org/2001/XMLSchema

WS-Addressing may be used with SOAP [*SOAP 1.2 Messaging Framework [p.18]* , *SOAP 1.1 [p.17]*] as described in Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing SOAP Binding [p.18]*]. WS-Addressing may be used with WSDL [*WSDL 2.0 Core Language [p.18]* , *WSDL 1.1 [p.18]*] described services as described in Web Services Addressing 1.0 - WSDL Binding[*WS-Addressing WSDL Binding [p.18]*]. Examples in this specification use an XML 1.0 [*XML 1.0 [p.17]*] representation but this is not a requirement.

All information items defined by this specification are identified by the XML namespace URI [*XML Namespaces [p.17]*] http://www.w3.org/2005/08/addressing. A normative XML Schema [*XML Schema Structures [p.17]* , *XML Schema Datatypes [p.17]*] document can be obtained by dereferencing the XML namespace URI.

2. Endpoint References

This section defines the information model and syntax of an endpoint reference.

This specification introduces the endpoint reference, a construct designed to support the following usage scenarios:

- Dynamic generation and customization of service endpoint descriptions.
- Referencing and description of specific service instances that are created as the result of stateful interactions.
- Flexible and dynamic exchange of endpoint information in tightly coupled environments where communicating parties share a set of common assumptions about specific policies or protocols that are used during the interaction.

2.1 Information Model for Endpoint References

An endpoint reference is a collection of abstract properties. This specification defines a core set of properties, but it is also possible for other specifications to extend these and/or add other properties. The semantics and XML Infoset representation for any such extension properties will be described in their defining specifications. An endpoint reference consists of the following abstract properties:

[address] : IRI (1..1)

An absolute IRI [*IETF RFC 3987 [p.17]*] representing the address of the endpoint. This specification introduces two predefined [address] values as shown in Table 2-1 [p.6] .

Table 2-1. Predefined [address] values

URI	**Description**
"http://www.w3.org/2005/08/addressing/anonymous"	Some endpoints cannot be located with a meaningful IRI; this URI is used to allow such endpoints to send and receive messages. The precise meaning of this URI is defined by the binding of Addressing to a specific protocol and/or the context in which the EPR is used.
"http://www.w3.org/2005/08/addressing/none"	Messages sent to EPRs whose [address] is this value MUST be discarded (i.e. not sent). This URI is typically used in EPRs that designate a reply or fault endpoint (see section **3.1 Abstract Property Definitions** [p.10]) to indicate that no reply or fault message should be sent.

[reference parameters] : xs:any (0..unbounded).

A reference may contain a number of individual parameters that are associated with the endpoint to facilitate a particular interaction. Reference parameters are namespace-qualified element information items that are required to properly interact with the endpoint. Reference parameters are provided by the issuer of the endpoint reference and are assumed to be opaque to other users of an endpoint reference. The binding of reference parameters to messages depends upon the protocol binding used to interact with the endpoint - Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing SOAP Binding [p.18]*] describes the default binding for the SOAP protocol.

The reference parameters are not ordered. No significance can be attributed to any order in which they may appear, because they may be bound to a message in a way which does not preserve that ordering.

[metadata] : xs:any (0..unbounded)

A reference may contain metadata that describes the behavior, policies and capabilities of the endpoint. Metadata may be included in an endpoint reference to facilitate easier processing by a user of an endpoint reference, or because the metadata was dynamically generated.

The metadata embedded in an EPR is not necessarily a complete statement of the metadata pertaining to the endpoint. Moreover, while embedded metadata is necessarily valid at the time the EPR is initially created it may become stale at a later point in time.

To deal with conflicts between the embedded metadata of two EPRs that have the same [address], or between embedded metadata and metadata obtained from a different source, or to ascertain the current validity of embedded metadata, mechanisms that are outside of the scope of this specification, such as EPR life cycle information (see **2.4 Endpoint Reference Lifecycle** [p.9]) or retrieval of metadata from an authoritative source, SHOULD be used.

2.2 Endpoint Reference XML Infoset Representation

This section defines an XML Infoset-based representation for an endpoint reference as both an XML type (wsa:EndpointReferenceType) and as an XML element (<wsa:EndpointReference>). For brevity simple XML terms are used, e.g. ’element’ instead of ’element information item’ - this is not intended to constrain use of the constructs defined in this section to textual XML representations.

The wsa:EndpointReferenceType type is used wherever a Web service endpoint is referenced. The following describes the contents of this type:

```
<wsa:EndpointReference>
    <wsa:Address>xs:anyURI</wsa:Address>
    <wsa:ReferenceParameters>xs:any*</wsa:ReferenceParameters> ?
    <wsa:Metadata>xs:any*</wsa:Metadata>?
</wsa:EndpointReference>
```

The following describes the attributes and elements listed in the schema overview above:

/wsa:EndpointReference

This represents some element of type wsa:EndpointReferenceType. This example uses the predefined <wsa:EndpointReference> element, but any element of type wsa:EndpointReferenceType may be used.

/wsa:EndpointReference/wsa:Address

This REQUIRED element (whose content is of type xs:anyURI) specifies the [address] property of the endpoint reference.

/wsa:EndpointReference/wsa:Address/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

/wsa:EndpointReference/wsa:ReferenceParameters

This OPTIONAL element may contain elements from any namespace. Such elements form the [reference parameters] of the reference.

/wsa:EndpointReference/wsa:ReferenceParameters/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

/wsa:EndpointReference/wsa:ReferenceParameters/{any}

Each element information item found in [reference parameters] (including all of that elements [children], [attributes] and [in-scope namespaces]) is represented as is.

/wsa:EndpointReference/wsa:Metadata

This OPTIONAL element may contain elements from any namespace. Such elements form the metadata that is relevant to the interaction with the endpoint.

/wsa:EndpointReference/wsa:Metadata/{any}

Each child element of Metadata represents an individual piece of metadata.

/wsa:EndpointReference/wsa:Metadata/{@any}

This is an extensibility mechanism to allow additional attributes to be specified. Some examples in this specification show use of this extensibility point to include a wsdlLocation[*WSDL 2.0 Core Language [p.18]*] attribute to provide a hint for the location of a WSDL description of the service deployed at the endpoint.

/wsa:EndpointReference/{any}

This is an extensibility mechanism to allow additional elements to be specified.

/wsa:EndpointReference/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

Note:

Specifications which describe any extension elements or attributes used to augment the above model will explain any effects those extensions may have on the abstract properties. They may affect either the core properties or extension properties as defined in **2.1 Information Model for Endpoint References** [p.6] .

The following shows an example endpoint reference. This element references the the endpoint at the URI "http://example.com/fabrikam/acct".

Example 2-1. Example endpoint reference.

```
<wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
   <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
</wsa:EndpointReference>
```

2.3 Endpoint Reference Comparison

This specification provides no concept of endpoint identity and therefore does not provide any mechanism to determine equality or inequality of EPRs and does not specify the consequences of their equality or inequality. However, note that it is possible for other specifications to provide a comparison function that is applicable within a limited scope.

2.4 Endpoint Reference Lifecycle

This specification does not define a lifecycle model for endpoint references and does not address the question of time-to-live for endpoint references. Other specifications that build on or use WS-Addressing may define a lifecycle model for endpoint references created according to that specification.

2.5 Endpoint Reference Extensibility

As noted in **2.2 Endpoint Reference XML Infoset Representation** [p.7] endpoint references are extensible. When extension attributes or elements appear as part of an endpoint reference, the processing model for such extensions is defined by the specification for those extensions. Software that processes endpoint references can safely ignore any such extensions that it does not recognize or understand.

Extension elements and attributes MAY add additional properties to an endpoint reference in addition to those specified in **2.1 Information Model for Endpoint References** [p.6] . Endpoint reference extensions MAY modify the value of one or more existing properties of an endpoint reference. Extensions MAY modify the rules for binding endpoint reference properties to message addressing properties, or otherwise indicate that a different binding be used.

Note that this ability to modify existing properties and binding behavior, when coupled with the fact that software can ignore unknown or unrecognized extensions, may result in a difference in behavior depending on whether such an extended endpoint reference is processed by software that understands the extension(s). When designing endpoint reference extensions designers should consider that standard processing per this specification will prevail in cases where their extension is not recognized or understood.

2.6 Identifying Resources in an Endpoint Reference

The Architecture of the World Wide Web, Volume One [*AoWWW [p.17]*] recommends [*AoWWW [p.17]* , Section 2] the use of URIs to identify resources. Using abstract properties of an EPR other than [destination] to identify resources is contrary to this recommendation. In certain circumstances, such a use of additional properties may be convenient or beneficial; however, when building systems, the benefits or convenience of identifying a resource using reference parameters should be carefully weighed against the benefits of identifying a resource solely by URI as explained in [*AoWWW [p.17]* , Section 2.1] of the Web

Architecture.

3. Message Addressing Properties

This section defines the information model and syntax of message addressing properties.

Message addressing properties provide references for the endpoints involved in an interaction. The use of these properties to support specific interactions is in general defined by both the semantics of the properties themselves and the implicit or explicit contract that governs the message exchange. If explicitly available, this contract can take different forms including but not being limited to WSDL MEPs and interfaces; business processes and e-commerce specifications, among others, can also be used to define explicit contracts between the parties.

In a one-way interaction pattern a source sends a message to a destination without any further definition of the interaction. "Request-response" is a common interaction pattern that consists of an initial message sent by a source endpoint (the request) and a subsequent message sent from the destination of the request back to the source (the response). A response in this case can be either an application message, a fault, or any other message. Note, however, that reply messages may be sent as part of other message exchanges as well, and are not restricted to the usual single Request, single Response pattern, or to a particular WSDL transmission primitive or MEP. The contract between the interacting parties may specify that multiple or even a variable number of replies be delivered.

The set of message addressing properties defined in this specification is sufficient for many simple variations of one-way and request-response MEPs. More advanced MEPs may require additional message addressing properties to augment the facilities provided here.

3.1 Abstract Property Definitions

Message addressing properties collectively augment a message with the following abstract properties to support one-way, request-response, and other interaction patterns:

[destination] : IRI (1..1)

An absolute IRI representing the address of the intended receiver of this message.

[source endpoint] : endpoint reference (0..1)

Reference to the endpoint from which the message originated.

[reply endpoint] : endpoint reference (0..1)

An endpoint reference for the intended receiver for replies to this message.

[fault endpoint] : endpoint reference (0..1)

An endpoint reference for the intended receiver for faults related to this message.

[action] : IRI (1..1)

An absolute IRI that uniquely identifies the semantics implied by this message.

It is RECOMMENDED that the value of the [action] property is an IRI identifying an input, output, or fault message within a WSDL interface or port type. An action may be explicitly or implicitly associated with the corresponding WSDL definition. Web Services Addressing 1.0 - WSDL Binding[*WS-Addressing WSDL Binding [p.18]*] describes the mechanisms of association.

[message id] : IRI (0..1)

An absolute IRI that uniquely identifies the message. When present, it is the responsibility of the sender to ensure that each message is uniquely identified. The behavior of a receiver when receiving a message that contains the same [message id] as a previously received message is unconstrained by this specification.

[relationship] : (IRI, IRI) (0..unbounded)

A pair of values that indicate how this message relates to another message. The type of the relationship is identified by an absolute IRI. The related message is identified by an absolute IRI that corresponds to the related message's [message id] property. The message identifier IRI may refer to a specific message, or be the following pre-defined URI that means "unspecified message": "http://www.w3.org/2005/08/addressing/unspecified"

This specification has one predefined relationship type as shown in Table 3-1 [p.11] .

Table 3-1. Predefined [relationship] values

URI	**Description**
"http://www.w3.org/2005/08/addressing/reply"	Indicates that this is a reply to the message identified by the [message id] IRI.

[reference parameters] : xs:any (0..unbounded).

Corresponds to the value of the [reference parameters] property of the endpoint reference to which the message is addressed.

The [destination] and [action] properties indicate the target processing location and the verb or intent of the message respectively. The values of these properties can be used to facilitate the dispatch of messages.

A binding of WS-Addressing message addressing properties MUST reflect the property cardinality shown above. Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing SOAP Binding [p.18]*] defines such a binding for the SOAP [*SOAP 1.2 Messaging Framework [p.18]* , *SOAP 1.1 [p.17]*] protocol.

3.2 XML Infoset Representation of Message Addressing Properties

The following shows the XML Infoset representation of the message addressing properties defined in **3.1 Abstract Property Definitions** [p.10] :

```
<wsa:To>xs:anyURI</wsa:To> ?
<wsa:From>wsa:EndpointReferenceType</wsa:From> ?
<wsa:ReplyTo>wsa:EndpointReferenceType</wsa:ReplyTo> ?
<wsa:FaultTo>wsa:EndpointReferenceType</wsa:FaultTo> ?
<wsa:Action>xs:anyURI</wsa:Action>
<wsa:MessageID>xs:anyURI</wsa:MessageID> ?
<wsa:RelatesTo RelationshipType="xs:anyURI"?>xs:anyURI</wsa:RelatesTo> *
<wsa:ReferenceParameters>xs:any*</wsa:ReferenceParameters> ?
```

The following describes the attributes and elements listed in the schema overview above:

/wsa:To

This OPTIONAL element (whose content is of type xs:anyURI) provides the value for the [destination] property. If this element is NOT present then the value of the [destination] property is "http://www.w3.org/2005/08/addressing/anonymous".

/wsa:From

This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [source endpoint] property.

/wsa:ReplyTo

This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [reply endpoint] property. If this element is NOT present then the value of the [address] property of the [reply endpoint] EPR is "http://www.w3.org/2005/08/addressing/anonymous".

/wsa:FaultTo

This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [fault endpoint] property.

/wsa:Action

This REQUIRED element (whose content is of type xs:anyURI) conveys the value of the [action] property.

/wsa:MessageID

This OPTIONAL element (whose content is of type xs:anyURI) conveys the [message id] property.

/wsa:RelatesTo

This OPTIONAL (repeating) element information item contributes one abstract [relationship] property value, in the form of an (IRI, IRI) pair. The content of this element (of type xs:anyURI) conveys the [message id] of the related message.

/wsa:RelatesTo/@RelationshipType

This OPTIONAL attribute (of type xs:anyURI) conveys the relationship type as an IRI. When absent, the implied value of this attribute is "http://www.w3.org/2005/08/addressing/reply".

/[reference parameters]*

Each element information item found in [reference parameters] (including all of that elements [children], [attributes] and [in-scope namespaces]) is represented as is.

Note that each of the element information items described above allows attribute wildcards for future extensibility. A message processor may safely ignore any extension attribute it does not recognize. Attribute extensibility allows use of xml:id[*xml:id [p.18]*] for identification of these elements if desired.

3.2.1 Comparing IRIs

The values of the Message Addressing Properties [action], [message id], and [relationship] are absolute IRIs. The purpose of these IRIs is primarily identification, rather than resource retrieval. As such, simple string comparison, as indicated in Internationalized Resource Identifiers *IETF RFC 3987 [p.17]* section 5.3.1, is sufficient to determine equivalence of these IRIs.

Comparison of [destination] property values is out of scope, other than using simple string comparison to detect whether the value is anonymous, that is, where [destination] has the value "http://www.w3.org/2005/08/addressing/anonymous".

3.3 Sending a Message to an EPR

This section describes the process of constructing a message in accordance to an EPR.

1. If the EPR's [address] property is "http://www.w3.org/2005/08/addressing/none" the message is discarded, if not then populate the message's message addressing properties:
    - [action]: this property is required, but is not populated from the EPR.
    - [destination]: this property takes the value of the EPR's [address] property.
    - [reference parameters]: this property takes the value of the selected EPR's [reference parameters] property

3.4 Formulating a Reply Message

This section specifies the WS-Addressing-specific rules for creating a reply or fault message related to another message.

1. Select the appropriate EPR:

   - If the reply is a normal message, select the EPR from the related message's [reply endpoint] message addressing property.

     Note:

     The [reply endpoint] message addressing property will always be present when using the XML Infoset representation since, in the absence of a wsa:ReplyTo element, the value of the [reply endpoint] message addressing property defaults to an EPR with an [address] property of "http://www.w3.org/2005/08/addressing/anonymous" - see section **3.2 XML Infoset Representation of Message Addressing Properties** [p.12] .

     If the [reply endpoint] message addressing property is not present the processor MUST fault. This could only occur when using an alternate representation of message addressing properties.

   - Otherwise, if the reply is a fault message and the related message's [fault endpoint] message addressing property is not empty, select the EPR from that property. If the [fault endpoint] property is empty, select the EPR from the related message's [reply endpoint] message addressing property. Otherwise, if the [reply endpoint] property is empty, the behavior of the recipient of the related message is unconstrained by this specification.

   - In either of the above cases, if the related message lacks a [message id] property, the processor MUST fault.

2. Send the message according to the previous section, but also including:

   - [relationship]: this property MUST include a pair of IRIs as follows; the relationship type is the predefined reply URI "http://www.w3.org/2005/08/addressing/reply" and the related message's identifier is the [message id] property value from the message being replied to; other relationships MAY be expressed in this property

The following example illustrates a message containing message addressing properties serialized as header blocks in a SOAP 1.2 message:

Example 3-1. Example message.

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
  <wsa:MessageID>http://example.com/someuniquestring</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://example.com/business/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To>mailto:fabrikam@example.com</wsa:To>
    <wsa:Action>http://example.com/fabrikam/mail/Delete</wsa:Action>
  </S:Header>
  <S:Body>
    <f:Delete xmlns:f="http://example.com/fabrikam">
```

```
        <maxCount>42</maxCount>
      </f:Delete>
   </S:Body>
</S:Envelope>
```

This message would have the following property values:

- [destination]: "mailto:fabrikam@example.com"
- [reply endpoint]: The endpoint with [address] "http://example.com/business/client1"
- [action]: "http://example.com/fabrikam/mail/Delete"
- [message id]: "http://example.com/someuniquestring"

The following example illustrates a reply to the above message:

Example 3-2. Example reply message.

```
<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>http://example.com/someotheruniquestring</wsa:MessageID>
    <wsa:RelatesTo>http://example.com/someuniquestring</wsa:RelatesTo>
    <wsa:To>http://example.com/business/client1</wsa:To>
    <wsa:Action>http://example.com/fabrikam/mail/DeleteAck</wsa:Action>
  </S:Header>
  <S:Body>
    <f:DeleteAck xmlns:f="http://example.com/fabrikam"/>
  </S:Body>
</S:Envelope>
```

This message would have the following property values:

- [destination]: "http://example.com/business/client1"
- [action]: "http://example.com/fabrikam/mail/DeleteAck"
- [message id]: "http://example.com/someotheruniquestring"
- [relationship]: ("http://www.w3.org/2005/08/addressing/reply", "http://example.com/someuniquestring")

4. Security Considerations

Conformance to this specification does not require a message receiver to honor the WS-Addressing constructs within a message if the receiver is not satisfied that the message is safe to process.

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. To an extent the content of such unsolicited messages can also be controlled using reference parameters supplied by the initial message sender. Because of these capabilities it is essential that communications using WS-Addressing are adequately secured and that a sufficient level of trust is established between the communicating parties before a receiver processes WS-Addressing constructs within a message. There are several aspects to securing a message:

1. EPRs and message addressing properties should be integrity-protected to prevent tampering. Such integrity protection might be provided by the transport, a message level signature, or use of an XML digital signature within EPRs.

2. Users of EPRs should validate the trustworthiness of an EPR before using it by considering the following aspects:

    1. whether the EPR was obtained from a trusted source

    2. whether the EPR was obtained from a source with authority to represent the [address] of that EPR

    3. whether the [address] of the EPR is a trusted destination

For example, the receiver of a message might rely on the presence of a verifiable signature by a trusted party over the message addressing properties to determine that the message originated from a trusted source and further require that the [reply endpoint] and [fault endpoint] are signed by a principle with authority to represent the [address] of those EPRs to ensure that unsolicted messages are not sent. Alternatively an out-of-band means of establishing trust might be used to determine whether a particular EPR is trustworthy.

4.1 Additional Security Considerations

To prevent information disclosure, EPR issuers should not put sensitive information into the [address] or [reference parameters] properties unless it has been adequately protected against arbitrary disclosure.

Some processors may use [message id] as part of a uniqueness metric in order to detect message replay. Care should be taken to ensure that, for purposes of replay detection, [message id] is composed from data, such as a timestamp, such that a legitimate retransmission of the message is not confused with a replay attack. It is also advisable to use a [message id] that is not predictable, to prevent attackers from constructing and sending an unsolicited reply to a message without having to see the actual message.

5. References

5.1 Normative References

[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.

[IETF RFC 3987]
Internationalized Resource Identifiers (IRIs) M. Duerst, and M. Suignard, Authors. 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. 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.

5.2 Other References

[AoWWW]
Architecture of the World Wide Web, Volume One, I. Jacobs, and N. Walsh, Editors. World Wide Web Consortium, 15 December 2004. This version of the Architecture of the World Wide Web, Volume One Recommendation is at http://www.w3.org/TR/2004/REC-webarch-20041215/. The latest version is available at http://www.w3.org/TR/webarch/

[SOAP 1.1]
Simple Object Access Protocol (SOAP) 1.1, D. Box, et al, Editors. World Wide Web Consortium, 8 May 2000. Available at http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

[SOAP 1.2 Messaging Framework]
SOAP Version 1.2 Part 1: Messaging Framework, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, and 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/.

[WS-Addressing SOAP Binding]
Web Services Addressing 1.0 - SOAP Binding, M. Gudgin, M. Hadley, and T. Rogers, Editors. World Wide Web Consortium, 9 May 2006. This version of the WS-Addressing SOAP Binding Recommendation is http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509. The latest version of WS-Addressing SOAP Binding is available at http://www.w3.org/TR/ws-addr-soap.

[WS-Addressing WSDL Binding]
Web Services Addressing 1.0 - WSDL Binding, M. Gudgin, M. Hadley, T. Rogers, and Ü. Yalçinalp, Editors. 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 1.1]
Web Services Description Language (WSDL) 1.1, E. Christensen, et al, Authors. World Wide Web Consortium, March 2001. Available at http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[WSDL 2.0 Core Language]
Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana, Editors. World Wide Web Consortium, 27 March 2006. This version of the WSDL 2.0 specification is http://www.w3.org/TR/2006/CR-wsdl20-20060327. The latest version of WSDL 2.0 is available at http://www.w3.org/TR/wsdl20.

[WS-Security 2004]
Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), A. Nadalin, C. Kaler, P. Hallam-Baker, and R. Monzillo, Editors. Organization for the Advancement of Structured Information Standards, March 2004. Available at http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

[xml:id]
xml:id Version 1.0, J. Marsh, D. Veillard, and N. Walsh, Editors. World Wide Web Consortium, 9 September 2005. This version of the xml:id Version 1.0 Recommendation is http://www.w3.org/TR/2005/REC-xml-id-20050909/. The latest version of xml:id is available at http://www.w3.org/TR/xml-id/.

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.), Alain Regnier (Ricoh Company, Ltd.), 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.