Web Services Addressing 1.0 - Core

W3C Working Draft 31 March 2005

This version:
http://www.w3.org/TR/2005/WD-ws-addr-core-20050331
Latest version:
http://www.w3.org/TR/ws-addr-core
Previous versions:
http://www.w3.org/TR/2005/WD-ws-addr-core-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

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.15]*] 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 Last Call Working Draft of the Web Services Addressing 1.0 - Core 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.

If the feedback is positive, the Working Group plans to submit this specification for consideration as a W3C Candidate Recommendation. Comments on this document are invited and are to be sent to the public public-ws-addressing-comments@w3.org mailing list (public archive). Comments can be sent until **11 May 2005**.

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.16] . Issues about this document are documented in the Last Call issues list maintained by the Working Group.

Discussion of this document takes place on the public-ws-addressing@w3.org mailing list (public archive).

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

1. Introduction [p.3]
   1.1 Notational Conventions [p.4]
   1.2 Namespaces [p.4]
2. Endpoint References [p.5]
   2.1 Information Model for Endpoint References [p.5]
   2.2 Endpoint Reference XML Infoset Representation [p.6]
   2.3 Endpoint Reference Comparison [p.7]
   2.4 Endpoint Reference Lifecycle [p.8]
   2.5 Endpoint Reference Extensibility [p.8]
3. Message Addressing Properties [p.8]
   3.1 XML Infoset Representation of Message Addressing Properties [p.11]
      3.1.1 Comparing IRIs [p.12]
   3.2 Formulating a Reply Message [p.12]
4. Security Considerations [p.14]
5. References [p.15]

Appendices

A. Acknowledgements [p.16] (Non-Normative)
B. Change Log [p.16] (Non-Normative)
  B.1 Changes Since Second Working Draft [p.16]
  B.2 Changes Since First Working Draft [p.17]
  B.3 Changes Since Submission [p.18]

---

1. Introduction

Web Services Addressing (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.

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 information headers.

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.

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
                xmlns:wsa="http://www.w3.org/2005/03/addressing">
(002)   <S:Header>
(003)    <wsa:MessageID>
(004)      http://example.com/6B29FC40-CA47-1067-B31D-00DD010662DA
(005)    </wsa:MessageID>
(006)    <wsa:ReplyTo>
(007)      <wsa:Address>http://example.com/business/client1</wsa:Address>
(008)    </wsa:ReplyTo>
(009)    <wsa:To>http://example.com/fabrikam/Purchasing</wsa:To>
(010)    <wsa:Action>http://example.com/fabrikam/SubmitPO</wsa:Action>
(011)   </S:Header>
(012)   <S:Body>
(013)     ...
(014)   </S:Body>
(015) </S:Envelope>
```

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

Lines (003) to (010) contain the message information header blocks. Specifically, lines (003) to (005) specify the identifier for this message and lines (006) to (008) specify the endpoint to which replies to this message should be sent as an Endpoint Reference. Line (009) specifies the address URI of the ultimate receiver of this message. Line (010) specifies an Action IRI 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.15]*].

When describing abstract data models, this specification uses the notational convention used by the XML Infoset [*XML Information Set [p.15]*]. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [*XML Schema Structures [p.15]* , *XML Schema Datatypes [p.15]*], this specification uses the notational convention of WS-Security [*WS-Security [p.16]*]. 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.15]*]).

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
xs	http://www.w3.org/2001/XMLSchema

WS-Addressing may be used with SOAP [*SOAP 1.2 Part 1: Messaging Framework [p.15]*] as described in Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing-SOAP [p.15]*]. WS-Addressing may be used with WSDL [*WSDL 2.0 [p.15]*] described services as described in Web Services Addressing 1.0 - WSDL Binding[*WS-Addressing-WSDL [p.15]*]. Examples in this specification use an XML 1.0 [*XML 1.0 [p.15]*] representation but this is not a requirement.

All information items defined by this specification are identified by the XML namespace URI [*XML Namespaces [p.15]*] "http://www.w3.org/2005/03/addressing". A normative XML Schema [*XML Schema Structures [p.15]* , *XML Schema Datatypes [p.15]*] 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 consists of the following abstract properties:

[address] : IRI (mandatory)

> An address IRI for the endpoint.

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

> A reference may contain a number of individual parameters which are associated with the endpoint to facilitate a particular interaction. Reference parameters are element information items that are named by QName and are required to properly interact with the endpoint. Reference parameters are also provided by the issuer of the endpoint reference and are otherwise assumed to be opaque to consuming applications. The use of reference parameters is dependent upon the protocol binding and data encoding used to interact with the endpoint. Web Services Addressing 1.0 - SOAP Binding[*WS-Addressing-SOAP [p.15]*] describes the default binding for the SOAP protocol.

[metadata] : xsd: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 the consuming application, or because the metadata was dynamically generated.
>
> The metadata embedded in each of the EPRs MAY differ, as the metadata carried by 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, 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 **2.4 Endpoint Reference Lifecycle** [p.8] 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>).

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

Example 2-1. Structure of the wsa:EndpointReference element.

```
<wsa:EndpointReference>
    <wsa:Address>xs:anyURI</wsa:Address>

    <wsa:ReferenceParameters>... </wsa:ReferenceParameters> ?
    <wsa:Metadata> ... </wsa:Metadata>?
    <xs:any/>*
</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 (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 contains the elements that convey 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 of found in [reference parameters] (including all of its [children], [attributes] and [in-scope namespaces]) is represented as is.

/wsa:EndpointReference/wsa:Metadata

This OPTIONAL element contains 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 [p.15]*] 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.

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

Example 2-2. Example endpoint reference.

```
<wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/03/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.6] 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 recognise 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.5] . 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 unrecognised extensions, may result in a difference in behaviour 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 whether they desire standard processing per this specification in cases where their extension is not recogonised or understood.

3. Message Addressing Properties

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

Editorial note	
The Working Group requests feedback regarding the mechanism for and description of Message Addressing Property extensibility beyond the MEPs currently described in the WSDL specifications, along with use cases that illustrate how referencing specifications and other users of Addressing intend to extend them. Although the Working Group has resolved upon a particular design, some participants believe it is not adequately specified. Such feedback will help the Working Group determine whether it needs to re-examine this issue.	

Message addressing properties provide references for the endpoints involved in an interaction. The use of these properties to support specific interaction 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 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.

The basic interaction pattern from which all others are composed is "one way". In this pattern a source sends a message to a destination without any further definition of the interaction. "Request Reply" 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 reply). A reply 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 Reply pattern, or to a particular WSDL MEP. The contract between the interacting parties may specify that multiple or even a variable number or replies be delivered.

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

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

[destination] : IRI (mandatory)

> 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. If a reply is expected, a message MUST contain a [reply endpoint]. The sender MUST use the contents of the [reply endpoint] to formulate the reply message as defined in **3.2 Formulating a Reply Message** [p.12] . If this property is present, the [message id] property is REQUIRED.

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

> An endpoint reference for the intended receiver for faults related to this message. When formulating a fault message as defined in **3.2 Formulating a Reply Message** [p.12] , the sender MUST use the contents of the [fault endpoint], when present, of the message being replied to to formulate the fault message. If this property is present, the [message id] property is REQUIRED.

[action] : IRI (mandatory)

> An identifier 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 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 [p.15]*] describes the mechanisms of association.

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

An IRI that uniquely identifies this message in time and space. No two messages with a distinct application intent may share a [message id] property. A message MAY be retransmitted for any purpose including communications failure and MAY use the same [message id] property. The value of this property is an opaque IRI whose interpretation beyond equivalence is not defined in this specification. If a reply is expected, this property MUST be present.

[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 IRI. The related message is identified by an 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 well-known IRI that means "unspecified message": "http://www.w3.org/2005/03/addressing/id/unspecified"

This specification has one predefined relationship type as shown in Table 3-1 [p.10] .

Table 3-1. Predefined [relationship] values

IRI	**Description**
"http://www.w3.org/2005/03/addressing/reply"	Indicates that this is a reply to the message identified by the IRI.

A reply message MUST contain a [relationship] property consisting of the predefined reply IRI and the message id property of the request message.

[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 dispatching of incoming messages is based on two message properties: the mandatory "destination" and "action" fields indicate the target processing location and the verb or intent of the message respectively.

Due to the range of network technologies currently in wide-spread use (e.g., NAT, DHCP, firewalls), many deployments cannot assign a meaningful global IRI to a given endpoint. To allow these "anonymous" endpoints to initiate message exchange patterns and receive replies, WS-Addressing defines the following well-known IRI for use by endpoints that cannot have a stable, resolvable IRI: "http://www.w3.org/2005/03/addressing/role/anonymous"

Requests whose [reply endpoint], [source endpoint] and/or [fault endpoint] use this address MUST provide some out-of-band mechanism for delivering replies or faults (e.g. returning the reply on the same transport connection). This mechanism may be a simple request/reply transport protocol (e.g., HTTP GET or POST). This IRI MAY be used as the [destination] for reply messages and SHOULD NOT be used as the [destination] in other circumstances.

3.1 XML Infoset Representation of Message Addressing Properties

Message addressing properties provide end-to-end characteristics of a message that can be easily secured as a unit. These properties are immutable and not intended to be modified along a message path.

The following describes the XML Infoset representation of message addressing properties:

Example 3-1. XML Infoset representation of message addressing properties.

```
<wsa:MessageID> xs:anyURI </wsa:MessageID>
<wsa:RelatesTo RelationshipType="..."?>xs:anyURI</wsa:RelatesTo>
<wsa:To>xs:anyURI</wsa:To>
<wsa:Action>xs:anyURI</wsa:Action>
<wsa:From>endpoint-reference</wsa:From>
<wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
<wsa:FaultTo>endpoint-reference</wsa:FaultTo>
```

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

/wsa:MessageID

This OPTIONAL element (of type xs:anyURI) conveys the [message id] property. This element MUST be present if wsa:ReplyTo or wsa:FaultTo is present.

/wsa:RelatesTo

This OPTIONAL (repeating) element information item contributes one abstract [relationship] property value, in the form of a (IRI, IRI) pair. The [children] property of this element (which is of type xs:anyURI) conveys the [message id] of the related message. This element MUST be present if the message is a reply.

/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/2005/03/addressing/reply".

/wsa:ReplyTo

This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [reply endpoint] property. This element MUST be present if a reply is expected. If this element is present, wsa:MessageID MUST be present.

/wsa:From

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

/wsa:FaultTo

This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [fault endpoint] property. If this element is present, wsa:MessageID MUST be present.

/wsa:To

This OPTIONAL element (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/2005/03/addressing/role/anonymous". Otherwise the [children] of this element convey the value of this property.

/wsa:Action

This REQUIRED element of type xs:anyURI conveys the [action] property. The [children] of this element convey the value of this property.

/[reference parameters]*

Each element information item of found in [reference parameters] (including all of its [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.

3.1.1 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.15]* section 5.3.1, is sufficient to determine equivalence of these IRIs.

3.2 Formulating a Reply Message

The reply to a WS-Addressing compliant request message MUST be compliant to WS-Addressing and is constructed according to the following rules:

1. Select the appropriate EPR:

    - If the reply is a normal message, select the EPR from the incoming message's [reply endpoint] message addressing property. If none is present, the processor MUST fault.

    - Otherwise, if the reply is a fault message and the incoming 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 incoming message's [reply endpoint] message addressing property. Otherwise, if the [reply endpoint] property is empty, the behavior of the recipient of the incoming message is unconstrained by this specification.

2. Populate the reply message's message addressing properties:

   - [destination]: this property takes the value of the selected EPR's [address] property

   - [relationship]: a new pair of IRIs is added to this value as follows; the relationship type is the predefined reply IRI "http://www.w3.org/2005/03/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

   - [reference parameters]: this property takes the value of the selected EPR's [reference parameters] property

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

Example 3-2. Example request message.

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/03/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 S:mustUnderstand="1">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 3-3. Example response message.

```
<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/03/addressing">
  <S:Header>
```

```
    <wsa:MessageID>
      http://example.com/someotheruniquestring
    </wsa:MessageID>
    <wsa:RelatesTo>
      http://example.com/someuniquestring
    </wsa:RelatesTo>
    <wsa:To S:mustUnderstand="1">
      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/2005/03/addressing/reply", "http://example.com/someuniquestring")

4. Security Considerations

Users of WS-Addressing and EPRs (i.e., entities creating, consuming or receiving Message Addressing Properties and EPRs) SHOULD only use EPRs from sources they trust. For example, such users might only use EPRs that are signed by parties the user of the EPR trusts, or have some out-of-band means of establishing trust.

EPRs and message addressing properties SHOULD be integrity protected to prevent tampering. Such optional integrity protection might be provided by transport, message level signature, and use of an XML digital signature within EPRs.

To prevent information disclosure, EPR issuers SHOULD NOT put sensitive information into the [address] or [reference parameters] properties.

Some processors may use message identifiers ([message id]) as part of a uniqueness metric in order to detect replays of messages. Care should be taken to ensure that for purposes of replay detection, the message identifier is combined with other data, such as a timestamp, so that a legitimate retransmission of the message is not confused with a replay attack.

5. References

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

[WS-Addressing-WSDL]
Web Services Addressing 1.0 - WSDL 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. Internet Engineering Task Force, June 1999. Available at http://www.ietf.org/rfc/rfc2119.txt.

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

[XML Schema Structures]
XML Schema Part 1: Structures 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/.

[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 Yalçınalp (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
2005-03-30 @ 21:02	plehegar	Removed some extra blanks Added the note from David Hull at http://lists.w3.org/Archives/Public/public-ws-addressing/2005Mar/0254.html per teleconference March 28, 2005
2005-03-21 @ 22:36	mgudgin	Incorporated resolution of issue 50 into Section 3.2
2005-03-21 @ 22:06	mgudgin	Updated with resolution to issue 54

2005-03-21 @ 20:47	mgudgin	Removed parenthetical statement '(and opaquely)' from description of [action] property in Section 3 per resolution on 2005-03-21 telcon
2005-03-21 @ 16:39	mgudgin	s/that value/that the value in description of [action] property in Section 3
2005-03-21 @ 16:37	mgudgin	Split paragraph 2 in Section 3 into two seperate paragraphs
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.
2005-03-02 @ 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/
2005-03-02 @ 14:07	mhadley	Added issue 52 resolution.
2005-02-28 @ 22:08	mhadley	Added resolution to issues 24 and 26
2005-02-27 @ 21:42	mhadley	Added issue 48 resolution
2005-02-27 @ 19:42	mhadley	Changed URI to IRI where appropriate.
2005-02-23 @ 14:34	mgudgin	Added new section 2.5: Endpoint Reference Extensibility per resolution of issue i042
2005-02-17 @ 16:16	mhadley	Added resolution to issue 44
2005-02-15 @ 22:53	mhadley	Added resolution to issue 46

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.
2005-01-23 @ 21:13	mgudgin	Incorporated resolution of issue i014; edits to Section 2.3
2005-01-23 @ 20:52	mgudgin	Incorporated resolution of issue i006; made wsa:To optional
2005-01-23 @ 19:32	mgudgin	Incorporated resolution of Issue i001 by removing Reference Properties
2005-01-17 @ 02:13	mgudgin	Incorporated Paco's proposal for resolving Issue 038
2005-01-16 @ 22:40	mgudgin	s/PortType/InterfaceName in certain examples
2004-12-17 @ 16:08	mhadley	Improved readability of introduction
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
2004-11-23 @ 21:38	mhadley	Updated titles of examples. Fixed table formatting and references. Replaced uuid URIs with http URIs in examples. Added document status.
2004-11-22 @ 15:40	mhadley	Removed reference to WS-Policy
2004-11-15 @ 19:43	mhadley	Fixed some inter and intra spec references.
2004-11-12 @ 21:19	mgudgin	Removed TBD sections
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 @ 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