

WS-Policy Attachment for WSDL 2.0

September 01, 2006

Version 1.0

Authors

Francisco Curbera, IBM
Kirill Gavrylyuk, Microsoft
Maryann Hondo, IBM
Anish Karmarkar, Oracle
Jonathan Marsh, Microsoft
Ashok Malhotra, Oracle
Anthony Nadalin, IBM
David Orchard, BEA Systems
Daniel Roth, Microsoft
Arthur Ryman, IBM
Jeffrey Schlimmer, Microsoft
Chris Sharp, IBM
Asir S Vedamuthu (Editor), Microsoft

Summary

WS-Policy Attachment for WSDL 2.0 is draft material intended as input to discussions in the WS-Policy Working Group at W3C. This material describes a mechanism for associating policy expressions with Web service constructs in WSDL 2.0, analogous to the existing feature in the WS-PolicyAttachment specification for associating policy expressions with Web service constructs in WSDL 1.1.

Contents

- 1. Introduction**
- 2. Example**
- 3. Attaching Policy Expressions**
- 4. Extension to WSDL Component Model**
- 5. Effective Policy**

1. Introduction

WS-Policy Attachment for WSDL 2.0 describes a mechanism for associating policy expressions with Web service constructs in WSDL 2.0. The mechanism consists of:

- A model for attaching policies to WSDL 2.0 constructs. The model defines:
 - A partitioning of WSDL constructs into service, endpoint, operation and message policy subjects.
 - The semantics of attaching a policy to each policy subject.
 - How to combine policies attached to more than one WSDL component within a single policy subject.

- o An XML representation of policy expressions attached to WSDL 2.0 constructs and annotating those policy expressions as required extensions using the WSDL-defined extensibility flag @wsdl20:required.

Appendix A lists all the XML Namespaces that are used in this document. (XML elements without a namespace prefix are from the Web Services Policy XML Namespace.)

2. Example

The example below illustrates the use of WS-Policy Attachment for WSDL 2.0:

```

<wsdl20:description>
...
  <Policy wsu:Id="common">
    <mtom:OptimizedMimeSerialization wsp:Optional="true"/>
    <wsap:UsingAddressing />
  </Policy>

  <Policy wsu:Id="secure">
    <ExactlyOne>
      <sp:TransportBinding>...</sp:TransportBinding>
      <sp:AsymmetricBinding>...</sp:AsymmetricBinding >
    </ExactlyOne>
  </Policy>

  <wsdl20:binding name="SecureBinding"
    interface="tns:RealTimeDataInterface" >
    <PolicyReference URI="#secure" />
    <wsdl20:operation name="GetRealQuote" >...</wsdl20:operation>
    ...
  </wsdl20:binding>

  <wsdl20:service name="RealTimeDataService"
    interface="tns:RealTimeDataInterface" >
    <wsdl20:endpoint name="RealTimeDataPort"
      binding="tns:SecureBinding">
      <PolicyReference URI="#common" />
      ...
    </wsdl20:endpoint>
  </wsdl20:service>
...
</wsdl20:description>

```

The `SecureBinding` WSDL binding description describes a binding for an interface that provides real-time quotes and book information on securities. (The prefixes `wsdl20` and `tns` are used here to denote the Web Services Description Language 2.0 XML Namespace and the target namespace of this WSDL document respectively.) To

require the use of security for these offerings, a policy expression that requires the use of either transport-level or message-level security is attached to the binding description. The policy expression applies to any message exchange associated with any endpoint that supports this binding description.

The `RealTimeDataPort` WSDL endpoint description describes an endpoint that supports the `SecureBinding` WSDL binding description. To require the use of addressing and allow the use of optimization (Optimized MIME Serialization as defined in the MTOM specification), a policy expression that represents the addressing requirement and optimization capability is attached to the endpoint description. The policy expression applies to any message exchange associated with the `RealTimeDataPort` endpoint.

In the above example, the `#secure` and `#common` policy expressions attached to the `SecureBinding` WSDL binding and `RealTimeDataPort` WSDL endpoint descriptions collectively apply to any message exchange associated with the `RealTimeDataPort` endpoint. The example below represents the combination of these two policies, that is, the effective policy for the `RealTimeDataPort` endpoint.

```
<Policy>
  <All>
    <Policy>
      <mtom:OptimizedMimeSerialization wsp:Optional="true"/>
      <wsap:UsingAddressing />
    </Policy>
    <Policy>
      <ExactlyOne>
        <sp:TransportBinding>...</sp:TransportBinding>
        <sp:AsymmetricBinding>...</sp:AsymmetricBinding >
      </ExactlyOne>
    </Policy>
  </All>
</Policy>
```

3. Attaching Policy Expressions

Policy attachment points in a WSDL 2.0 document are:

- o `wSDL20:service`
- o `wSDL20:endpoint`
- o `wSDL20:binding`
- o `wSDL20:binding/wSDL20:operation`
- o `wSDL20:binding/wSDL20:fault`
- o `wSDL20:binding/wSDL20:operation/wSDL20:input`
- o `wSDL20:binding/wSDL20:operation/wSDL20:output`
- o `wSDL20:binding/wSDL20:operation/wSDL20:inFault`
- o `wSDL20:binding/wSDL20:operation/wSDL20:outFault`
- o `wSDL20:interface`

- o `wsdl20:interface/wsdl20:operation`
- o `wsdl20:interface/wsdl20:fault`
- o `wsdl20:interface/wsdl20:operation/wsdl20:input`
- o `wsdl20:interface/wsdl20:operation/wsdl20:output`
- o `wsdl20:interface/wsdl20:operation/wsdl20:infault` and
- o `wsdl20:interface/wsdl20:operation/wsdl20:outfault`.

Any of these elements may have one or more `Policy` or `PolicyReference` child elements.

Policy attachment points in a WSDL document are associated with specific policy subjects as described in the table below. There are four policy subjects in WSDL: the service policy subject, the endpoint policy subject, the operation policy subject and the message policy subject. When a policy expression is attached to a policy subject in a WSDL document, capabilities and requirements represented by the policy expression apply to any message exchange or message associated with (or described by) the policy subject.

Association of Policy Attachment Points with Policy Subjects

<i>Policy Attachment Point in a WSDL document</i>	<i>WSDL Component</i>	<i>Policy Subject</i>
<code>wsdl20:service</code>	Service	Service
<code>wsdl20:endpoint</code>	Endpoint	Endpoint
<code>wsdl20:binding</code>	Binding	
<code>wsdl20:interface</code>	Interface	
<code>wsdl20:binding/wsdl20:operation</code>	Binding Operation	Operation
<code>wsdl20:interface/wsdl20:operation</code>	Interface Operation	
<code>wsdl20:binding/wsdl20:operation/wsdl20:input</code>	Binding Message Reference	Message for an input message
<code>wsdl20:interface/wsdl20:operation/wsdl20:input</code>	Interface Message Reference whose {direction} property is 'in'	
<code>wsdl20:binding/wsdl20:operation/wsdl20:output</code>	Binding Message Reference	Message for an output message
<code>wsdl20:interface/wsdl20:operation/wsdl20:output</code>	Interface Message Reference whose {direction} property is 'out'	
<code>wsdl20:binding/wsdl20:fault</code>	Binding Fault	Message for an input fault message
<code>wsdl20:binding/wsdl20:operation/wsdl20:infault</code>	Binding Fault Reference	
<code>wsdl20:interface/wsdl20:fault</code>	Interface Fault	

wsd120:interface/wsd120:operation/ wsdl120:infault	Interface Fault Reference whose {direction} property is 'in'	
wsd120:binding/wsd120:fault	Binding Fault	Message for an output fault message
wsd120:binding/wsd120:operation/ wsdl120:outfault	Binding Fault Reference	
wsd120:interface/wsd120:fault	Interface Fault	
wsd120:interface/wsd120:operation/ wsdl120:outfault	Interface Fault Reference whose {direction} property is 'out'	

For a WSDL component, the attached policy (extension to the WSDL component model is described in [Section 4](#)) is considered an intrinsic part of the WSDL component definition and applies to all uses of that definition. For example, when attached to a WSDL Interface component, capabilities and requirements represented by a policy apply to all the use of this WSDL Interface description. When attached to a WSDL Binding component, capabilities and requirements represented by a policy apply to all the Endpoints that support this binding description.

A policy associated with a service policy subject applies to any message exchange (that is explicitly described by the Interface component in the Service component's {interface} property) using any of the endpoints offered by that service.

Policies associated with an endpoint policy subject apply to any message exchange (that is explicitly described by the Interface component in the Service component's {interface} property of the Endpoint component's {parent} property) made using that endpoint. Given that a WSDL Interface component may be used by one or more binding descriptions, it is recommended that only a policy containing policy assertions that apply to any possible binding description should be attached.

Policies associated with an operation policy subject apply to the message exchange described by that operation. Given that a WSDL Interface Operation component may be used by one or more binding descriptions, it is recommended that only a policy containing policy assertions that apply to any possible binding description should be attached.

Policies associated with a message policy subject apply to that message (input, output or fault). Given that a WSDL Interface Message Reference, Interface Fault, Interface Fault Reference components may be used by one or more binding descriptions, it is recommended that only a policy containing policy assertions that apply to any possible binding should be attached.

Policies may be attached at different levels of the WSDL component hierarchy. A message exchange with an endpoint may be described by the policies in all four policy subjects simultaneously.

The common mechanism of associating a policy expression with a policy subject is to attach a reference to the policy expression to the policy subject. As described in the WS-Policy specification, a reference to a policy expression is represented using the `PolicyReference` element. A policy attachment to a WSDL element is represented by attaching a `PolicyReference` element as a child element of the WSDL element.

Policy expressions can be included within a WSDL document or may reside external to a WSDL document. If including policy expressions with a WSDL document is the chosen approach, it is recommended that the `Policy` elements are included as children of the `wsdl20:description` element after the `wsdl20:types` element and referenced using the `PolicyReference` elements.

To mandate the processing of a policy expression attached to a policy attachment point in a WSDL document, the expression must be marked as required using the `@wsdl20:required` flag.

If the `Policy` elements are included as children of the `wsdl20:description` element, these `Policy` elements must not be marked as required using the `@wsdl20:required`. (Note: these expressions may be included as children of the `wsdl20:description` element and may not be attached to any policy attachment point in a WSDL document.)

4. Extension to WSDL Component Model

This document adds an optional `{policy}` property to the following WSDL components:

- Service
- Endpoint
- Binding
- Binding Operation
- Binding Fault
- Binding Message Reference
- Binding Fault Reference
- Interface
- Interface Operation
- Interface Fault
- Interface Message Reference
- Interface Fault Reference

The `{policy}` property, when present, represents the capabilities and requirements as a policy. The value of the `{policy}` property is a policy as defined by Section 3 – Policy Model in the WS-Policy specification. The following table describes the mapping from XML representation to the `{policy}` property.

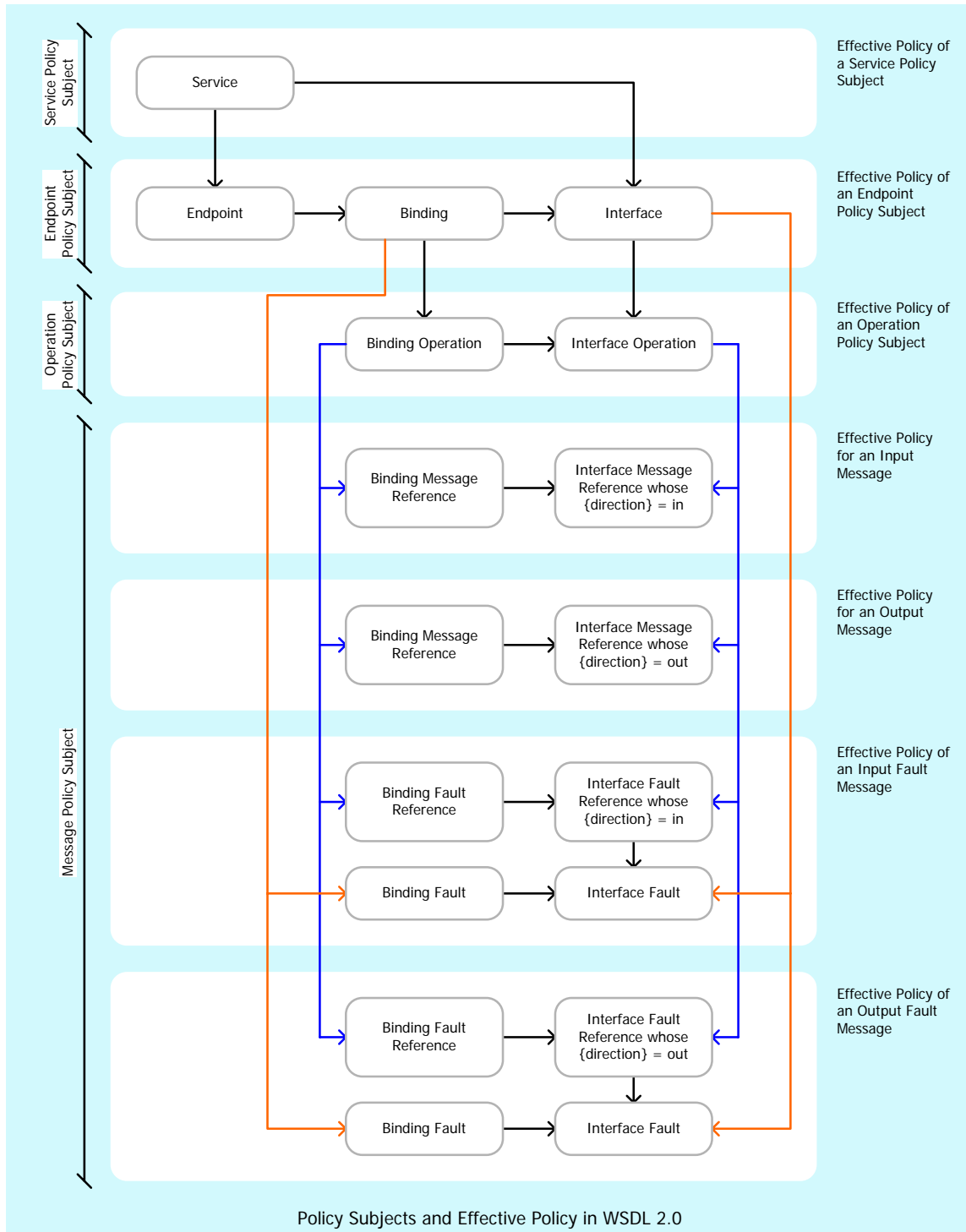
Mapping from XML representation to WSDL component properties

<i>Component</i>	<i>Property</i>	<i>Value</i>
Service	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:service</code> element.
Endpoint	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:endpoint</code> element.
Binding	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> element, if any, in the [children] of the <code>wsdl20:binding</code> elements.
Binding Operation	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:binding/wsdl20:operation</code> element.
Binding Fault	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:binding/wsdl20:fault</code> element.
Binding Message Reference	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:binding/wsdl20:operation/wsdl20:input</code> or <code>wsdl20:binding/wsdl20:operation/wsdl20:output</code> element.
Binding Fault Reference	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:binding/wsdl20:operation/wsdl20:infaul</code> or <code>wsdl20:binding/wsdl20:operation/wsdl20:outfaul</code> element.
Interface	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:interface</code> element.
Interface Operation	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:interface/wsdl20:operation</code> element, if any.
Interface Fault	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:interface/wsdl20:fault</code> element.
Interface Message Reference	{policy}	A policy corresponding to the merge of <code>Policy</code> or <code>PolicyReference</code> elements, if any, in the [children] of the <code>wsdl20:interface/wsdl20:operation/wsdl20:input</code> or <code>wsdl20:interface/wsdl20:operation/wsdl20:output</code> element.
Interface	{policy}	A policy corresponding to the merge of <code>Policy</code> or

Fault Reference		PolicyReference elements, if any, in the [children] of the wsdl20:interface/wsdl20:operation/wsdl20:infault or wsdl20:interface/wsdl20:operation/wsdl20:outfault element.
-----------------	--	---

5. Effective Policy

The following diagram illustrates the four policy subjects in WSDL and how the effective policy is calculated for each of these policy subjects.



Policy Subjects and Effective Policy in WSDL 2.0

If multiple policies are attached to WSDL components that collectively represent a policy subject then the effective policy of these policies applies. (For example, there is a policy attached to an Endpoint component that describes the component and there is a policy attached to the Binding component in the Endpoint component's {binding} property.) The effective policy is the merge of the policies that are

attached to the same policy subject. The rest of this section describes how the effective policy is calculated for each of these policy subjects.

Service Policy Subject

The effective policy of a service policy subject is the policy in the {policy} property of a Service component that describes the service.

Endpoint Policy Subject

The effective policy of an endpoint policy subject is the merge of policies in the {policy} properties of:

- An Endpoint component that describes the endpoint,
- The Binding component in the Endpoint component's {binding} property,
- The Interface component in the Service component's {interface} property of the Endpoint component's {parent} property and
- The Interface components in the {extended interfaces} property of the Interface component in the Service component's {interface} property of the Endpoint component's {parent} property.

Operation Policy Subject

If the Binding component has an Interface component in the {interface} property, then the effective policy of an operation policy subject may be calculated by merging the policies in the {policy} properties of:

- The Interface Operation component that describes the operation and
- The Binding Operation component (if any) whose {interface operation} property has the Interface Operation component.

If the Binding component does not have an Interface component in the {interface} property, then the effective policy of an operation policy subject must be calculated in relation to a specific endpoint, and is the policy in the {policy} property of the Interface Operation component that describes the operation.

Message Policy Subject (input message)

If the Binding component has an Interface component in the {interface} property, then the effective policy of an input message may be calculated by merging the policies in the {policy} properties of:

- The Interface Message Reference component that describes the input message and
- The Binding Message Reference component whose {interface message reference} property has the Interface Message Reference component.

If the Binding component does not have an Interface component in the {interface} property, then the effective policy of an input message must be calculated in relation to a specific endpoint, and is the policy in the {policy} property of the Interface Message Reference component that describes the input message.

Message Policy Subject (output message)

If the Binding component has an Interface component in the {interface} property, then the effective policy of an output message may be calculated by merging the policies in the {policy} properties of:

- The Interface Message Reference component that describes the output message and
- The Binding Message Reference component whose {interface message reference} property has the Interface Message Reference component.

If the Binding component does not have an Interface component in the {interface} property, then the effective policy of an output message must be calculated in relation to a specific endpoint, and is the policy in the {policy} property of the Interface Message Reference component that describes the output message.

Message Policy Subject (input fault message)

If the Binding component has an Interface component in the {interface} property, then the effective policy of an input fault message may be calculated by merging the policies in the {policy} properties of:

- The Interface Fault Reference component that describes the input fault message,
- The Interface Fault component in the Interface Fault Reference component's {interface fault} property,
- The Binding Fault Reference component whose {interface fault reference} property has the Interface Fault Reference component and
- The Binding Fault component whose {interface fault} property has the Interface Fault component in the Interface Fault Reference component's {interface fault} property.

If the Binding component does not have an Interface component in the {interface} property, then the effective policy of an input fault message must be calculated in relation to a specific endpoint, and is the merge of policies in the {policy} properties of:

- The Interface Fault Reference component that describes the input fault message and
- The Interface Fault component in the Interface Fault Reference component's {interface fault} property.

Message Policy Subject (output fault message)

If the Binding component has an Interface component in the {interface} property, then the effective policy of an output fault message may be calculated by merging the policies in the {policy} properties of:

- The Interface Fault Reference component that describes the output fault message,
- The Interface Fault component in the Interface Fault Reference component's {interface fault} property,
- The Binding Fault Reference component whose {interface fault reference} property has the Interface Fault Reference component and
- The Binding Fault component whose {interface fault} property has the Interface Fault component in the Interface Fault Reference component's {interface fault} property for the endpoint.

If the Binding component does not have an Interface component in the {interface} property, then the effective policy of an output fault message must be calculated in relation to a specific endpoint, and is the merge of policies in the {policy} properties of:

- o The Interface Fault Reference component that describes the output fault message and
- o The Interface Fault component in the Interface Fault Reference component's {interface fault} property.

Appendix A – XML Namespaces

[Table 1](#) lists XML namespaces that are used in this document. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 1: Prefixes and XML Namespaces used in this specification.

Prefix	XML Namespace	Specification(s)
mtom	http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization	[WS-OptimizedSerializationPolicy]
sp	http://schemas.xmlsoap.org/ws/2005/07/securitypolicy	[WS-SecurityPolicy]
wsa	http://www.w3.org/2005/08/addressing	[WS-Addressing]
wsap	http://www.w3.org/2006/05/addressing/wSDL	[WS-AddressingPolicy]
wSDL20	http://www.w3.org/2006/01/wSDL	[WSDL 2.0]
(none), wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	[WS-Policy , WS-PolicyAttachment]
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	[WS-Security 2004]

Appendix B – Normative References

[WSDL 2.0]

R. Chinnici, et al, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language," March 2006. (See <http://www.w3.org/TR/2006/CR-wsdl20-20060327/>.)

[WS-Policy]

S. Bajaj, et al, "Web Services Policy Framework (WS-Policy)," March 2006. (See <http://schemas.xmlsoap.org/ws/2004/09/policy>.)

[WS-PolicyAttachment]

S. Bajaj, et al, "Web Services Policy Attachment (WS-PolicyAttachment)," March 2006. (See <http://schemas.xmlsoap.org/ws/2004/09/policy>.)

Appendix C – Informative References

[MTOM]

M. Gudgin, et al, "SOAP Message Transmission Optimization," January 2005. (See <http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>.)

[Understanding WS-Policy]

A. Vedamuthu, et al, "Understanding Web Services Policy," July 2006. (See <http://msdn.microsoft.com/ws/2006/07/understanding-ws-policy/>.)

[WS-Addressing]

M. Gudgin, et al, "Web Services Addressing 1.0 - Core," May 2006. (See <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>.)

[WS-AddressingPolicy]

M. Gudgin, et al, "Web Services Addressing 1.0 – WSDL Binding," May 2006. (See <http://www.w3.org/TR/2006/CR-ws-addr-wsdl-20060529/>. The latest version of this specification is available at <http://www.w3.org/TR/ws-addr-wsdl/>.)

[WS-Security 2004]

A. Nadalin, et al, "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)," March 2004. (See <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>.)

[WS-SecurityPolicy]

G. Della-Libera, et al, "Web Services Security Policy Language (WS-SecurityPolicy)," July 2005. (See <http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/>.)

[XOP]

M. Gudgin, et al, "XML-binary Optimized Packaging," January 2005. (See <http://www.w3.org/TR/2005/REC-xop10-20050125/>.)