

Web Services Policy Assertions Language (WS-PolicyAssertions)

Version 1.1
28 May 2003

Authors

Don Box, Microsoft
Maryann Hondo, IBM
Chris Kaler, Microsoft
Hiroshi Maruyama, IBM
Anthony Nadalin (Editor), IBM
Nataraj Nagaratnam, IBM
Paul Patrick, BEA
Claus von Riegen, SAP
John Shewchuk, Microsoft

Copyright Notice

(c) 2001, 2002, 2003 BEA Systems, Inc, International Business Machines Corporation, Microsoft Corporation, SAP AG. All rights reserved.

BEA, IBM Microsoft, and SAP (collectively, the "Authors") hereby grant you permission to copy and display the WS-PolicyAssertions Specification, in any medium without fee or royalty, provided that you include the following on ALL copies of the WS-PolicyAssertions Specification, or portions thereof, that you make:

1. A link or URL to the Specification at this location
2. The copyright notice as shown in the WS-PolicyAssertions Specification.

EXCEPT FOR THE COPYRIGHT LICENSE GRANTED ABOVE, THE AUTHORS DO NOT GRANT, EITHER EXPRESSLY OR IMPLIEDLY, A LICENSE TO ANY INTELLECTUAL PROPERTY, INCLUDING PATENTS, THEY OWN OR CONTROL.

THE WS-POLICYASSERTIONS SPECIFICATION IS PROVIDED "AS IS," AND THE AUTHORS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE WS-POLICYASSERTIONS SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE WS-POLICYASSERTIONS SPECIFICATION.

The WS-PolicyAssertions Specification may change before final release and you are cautioned against relying on the content of this specification.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to the Specification or its contents without specific, written prior permission. Title to copyright in the WS-PolicyAssertions Specification will at all times remain with the Authors.

No other rights are granted by implication, estoppel or otherwise.

Abstract

This document specifies a set of common message policy assertions that can be specified within a policy.

Composable Architecture

By using the XML, SOAP, and WSDL extensibility models, the WS* specifications are designed to be composed with each other to provide a rich Web services environment. WS-PolicyAssertions by itself does not provide a negotiation solution for Web services. WS-PolicyAssertions is a building block that is used in conjunction with other Web service and application-specific protocols to accommodate a wide variety of policy exchange models.

Status

This WS-PolicyAssertions Specification is an initial public draft release and is provided for review and evaluation only. BEA, IBM, Microsoft, and SAP hope to solicit your contributions and suggestions in the near future. BEA, IBM, Microsoft, and SAP make no warranties or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction

1.1 Goals

1.2 Example

2. Notations and Terminology

2.1 Notational Conventions

2.2 Namespaces

2.3 Schema Files

2.4 Terminology

3. General Assertions

3.1 TextEncoding Assertion

3.2 Language Assertion

3.3 SpecVersion Assertion

3.4 MessagePredicate Assertion

4. Security Considerations

5. Acknowledgements

6. References

Appendix I. XPath Expressions for Policies

Appendix II. Message Part Selection Functions

1. Introduction

This specification defines general messaging-related assertions for use with WS-Policy [[WS-Policy](#)].

1.1 Goals

The goal of WS-PolicyAssertions is to provide an initial set of assertions to address some common needs of Web Services applications.

1.2 Example

The following example illustrates a policy expression:

```
001 <wsp:Policy xmlns:x="...">
002   <wsp:All>
003     <wsp:TextEncoding wsp:Usage="wsp:Required"
004       Encoding="iso-8859-5" />
005     <wsp:SpecVersion wsp:Usage="wsp:Required"
006       URI="http://schemas.xmlsoap.org/ws/2002/12/policy" />
007     <wsp:ExactlyOne>
008       <wsp:Language wsp:Usage="wsp:Required"
009         wsp:Preference="10" Language="da" />
010       <wsp:Language wsp:Usage="wsp:Required"
011         wsp:Preference="7" Language="en-gb" />
012       <wsp:Language wsp:Usage="wsp:Required"
013         wsp:Preference="1" Language="en" />
014     </wsp:ExactlyOne>
015   </wsp:All>
016 </wsp:Policy>
```

In this example, we illustrate the expression of general policies. Line 3 represents a specific policy assertion for text encoding.

Line 4 illustrates the assertion of a reference to a specification.

Lines 5-9 represent the `<wsp:ExactlyOne>` policy operator that is used to produce policy assertion sets. That is, a valid policy must contain one and only one of the contained assertions (Lines 6-8).

2. Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

2.1 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 [[RFC 2119](#)].

Namespace [[XML-NS](#)] URIs (of the general form "some-URI") represents some application-dependent or context-dependent URI as defined in RFC 2396 [[RFC 2396](#)].

2.2 Namespaces

The XML namespace URI that MUST be used by implementations of this specification is:

<http://schemas.xmlsoap.org/ws/2002/12/policy>

A normative copy of the XML Schema for WS-PolicyAssertions constructs may be retrieved by resolving the URI "<http://schemas.xmlsoap.org/ws/2002/12/policy>".

The following namespaces are used in this document:

Prefix	Namespace
S	http://www.w3.org/2002/06/soap-envelope
xsd	http://www.w3.org/2001/XMLSchema
wsse	http://schemas.xmlsoap.org/ws/2002/12/secext
wsp	http://schemas.xmlsoap.org/ws/2002/12/policy
wsu	http://schemas.xmlsoap.org/ws/2002/07/utility

2.3 Schema Files

The XML Schema [[XMLSchema1](#)] for this specification can be located at:

<http://schemas.xmlsoap.org/ws/2002/12/policy>

(See also [Appendix III](#).)

In this document reference is made to the *wsu:Id* attribute and the `<wsu:Created>` and `<wsu:Expires>` elements in a utility schema (<http://schemas.xmlsoap.org/ws/2002/07/utility>). The *wsu:Id* attribute and the `<wsu:Created>` and `<wsu:Expires>` elements were added to the utility schema with the intent that other specifications requiring such an ID or timestamp could reference it (as is done here).

2.4 Terminology

We introduce the following terms which are used throughout this document:

Policy – A *policy* is a set of domain-specific policy assertions.

Policy Assertion – A *policy assertion* represents an individual preference, requirement, capability or other property.

3. General Assertions

This section describes a set of policy assertions that MAY be used within policy expressions. Unless otherwise noted these assertions may be used by both the Web service and the client.

WS-Policy [[WS-Policy](#)] identifies the characteristics of a valid policy expression. For example, a usage must be specified for assertions. When combining the assertions identified here with other assertions, the rules defined in WS-Policy must be followed.

3.1 TextEncoding Assertion

A Web service may support more than one character set for encoding textual data (including XML 1.0). For example, *ISO-8859-1* or *ISO-8859-5* or *UTF-8*. In such cases, a mechanism is required to indicate which character encodings are supported. The `<wsp:TextEncoding>` element is used to indicate this.

This element uses syntax described in Section 4.3.3 of XML 1.0 (the EncName production).

The schema outline for `<wsp:TextEncoding>` is as follows:

```
<wsp:TextEncoding Encoding="..." wsp:Usage="..."? wsp:Preference="..."? >
  ...
</wsp:TextEncoding>
```

The following describes the attributes and tags listed in the schema outlined above:

/TextEncoding

This indicates supported character sets.

/TextEncoding/@Encoding

The value is specified as defined in Section 4.3.3 of XML 1.0 Second Edition [[XML](#)].

/TextEncoding/@wsp: Usage

This attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy [[WS-Policy](#)].

/TextEncoding/@wsp: Preference

This attribute specifies the preference of this particular alternative. The preference is expressed as an `xsd:int` [[XMLSchema2](#)]. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/TextEncoding/{any}

Additional child elements MAY be specified but MUST NOT contradict the semantics of the parent element; if an element is not recognized, it SHOULD be ignored.

/TextEncoding/@{any}

Additional attributes MAY be specified but MUST NOT contradict the semantics of the owner element; if an attribute is not recognized, it SHOULD be ignored.

The following example illustrates the use of this declaration:

```
<wsp:TextEncoding wsp:Usage="wsp:Required" Encoding="iso-8859-5" />
```

3.2 Language Assertion

A Web service may allow alternative natural languages to be used in message content. Consequently, there is a need for Web services to express their supported and preferred languages. The `<wsp:Language>` element is used to do this.

Note that the scope of usage of the `<wsp:Language>` element corresponds to the `xml:lang` attribute defined in the XML 1.0 Second Edition specification [[XML](#)]. It is expected that additional specifications will address internationalization preferences and capabilities in a distributed system.

The schema outline for `<wsp:Language>` is as follows:

```
<wsp:Language wsp:Usage="..."? wsp:Preference="..."? Language="..." >
  ...
</wsp:Language>
```

The following describes the attributes and tags listed in the schema outlined above:

/Language

This indicates supported languages.

/Language/@Language

The language value as specified in Section 2.12 of XML 1.0 Second Edition [[XML](#)].

/Language/@wsp:Usage

This attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy [[WS-Policy](#)].

/Language/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an `xsd:int` [[XMLSchema2](#)]. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/Language/{any}

Additional child elements MAY be specified but MUST NOT contradict the semantics of the parent element; if an element is not recognized, it SHOULD be ignored.

/Language/@{any}

Additional attributes MAY be specified but MUST NOT contradict the semantics of the owner element; if an attribute is not recognized, it SHOULD be ignored.

The following example illustrates the use of this declaration. This example indicates the policy prefers Danish, but will accept British English and other types of English.

```
<wsp:ExactlyOne>
  <wsp:Language wsp:Usage="wsp:Required"
    wsp:Preference="10"
    Language="da" />
  <wsp:Language wsp:Usage="wsp:Required"
    wsp:Preference="7"
    Language="en-gb" />
  <wsp:Language wsp:Usage="wsp:Required"
    wsp:Preference="1"
    Language="en" />
</wsp:ExactlyOne>
```

3.3 SpecVersion Assertion

Over time different specifications are revised. As a result, Web services may support different versions of the same specification. In some cases the specifications will be backwardly compatible; in others they won't. As a result, Web services need a way to indicate which specifications, and which versions of those specifications, they support. The `<wsp:SpecVersion>` element is used to indicate this.

The schema outline for `<wsp:SpecVersion>` is as follows:

```
<wsp:SpecVersion wsp:Usage="..."?
  URI="..."
  wsp:Preference="..."? >
</wsp:SpecVersion>
```

The following describes the attributes and tags listed in the schema outlined above:

/SpecVersion

This identifies compliance with a specific version of a specification. (Compliance is defined by the referenced specification.) There may be multiple specification elements to indicate compliance with multiple versions of the specification.

/SpecVersion/@wsp:Usage

This attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy [[WS-Policy](#)].

/SpecVersion/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an `xsd:int` [[XMLSchema2](#)]. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/SpecVersion/@URI

This is the URI of the specification; the URI SHOULD refer to a specific version of a specification using the specification URI designed by the organization creating the specification. For example: The WC3 SOAP/1.1 specification [[SOAP11](#)] is identified by the URI "http://www.w3.org/TR/2000/NOTE-SOAP-20000508/". The June 2002 working draft of SOAP/1.2 [[SOAP12](#)] is identified by the URI "http://www.w3.org/TR/2002/WD-soap12-part1-20020626/".

/SpecVersion/{any}

Additional child elements MAY be specified but MUST NOT contradict the semantics of the parent element; if an element is not recognized, it SHOULD be ignored.

/SpecVersion/@{any}

Additional attributes MAY be specified but MUST NOT contradict the semantics of the owner element; if an attribute is not recognized; it SHOULD be ignored.

The following example illustrates the use of this declaration:

```
<wsp:SpecVersion wsp:Usage="wsp:Required"
                URI="http://schemas.xmlsoap.org/ws/2002/12/policy" />
```

In some cases, a specification has multiple parts and just specifying the version of the specification isn't enough. In such cases, the specification outlines the different classes of support as elements in its schema. These tags, as appropriate, can be specified within the `<SpecVersion>` element. The following example illustrates support for "part 1" of the "framitz" specification whose schema is dated 12/2001.

```
<wsp:SpecVersion xmlns=".../utility"
                wsp:Usage="wsp:Required"
                URI="http://www.fabrikam123.com/2001/12/framitz">
  <f:part1 xmlns:f="..." />
</wsp:SpecVersion>
```

3.4 MessagePredicate Assertion

In this section we provide a policy assertion that MAY be used to ensure that a message conforms to a given predicate (pre-condition). For example, some message parts may be optional per a specification, but required by a specific implementation. The `<wsp:MessagePredicate>` element is used to indicate this.

The schema outline for this element is as follows:

```
<wsp:MessagePredicate wsp:Usage="..."? Dialect="..."? >
  ...
</wsp:MessagePredicate>
```

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

/MessagePredicate

The contents of this element (of type `xsd:string` [[XMLSchema2](#)]) specify a predicate expression that must evaluate to true for all conformant messages. The evaluation of the expression is determined by the optional dialect attribute. The default dialect is "<http://www.w3.org/TR/1999/REC-xpath-19991116>" indicating the expression is an XPath 1.0 expression [[XPATH](#)]. If there are multiple `<MessagePredicate>` elements specified, all must be true unless they are contained in a choice policy element. (See WS-Policy [[WS-Policy](#)].) If the selection of the targets can be easily expressed using the "<http://schemas.xmlsoap.org/2002/12/wsse#part>" dialect, then it is RECOMMENDED. Otherwise it is RECOMMENDED that a general XPath expressions using the functions defined in [Appendix I](#) be used.

/MessagePredicate/@wsp:Usage

This attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/MessagePredicate/@Dialect

The optional attribute identifies the expression dialect in use as a URI. If the attribute is not present, then XPath 1.0 is assumed.

URI	Meaning
http://www.w3.org/TR/1999/REC-xpath-19991116 (Default)	The contents of the <code><MessagePredicate></code> element is an XPath 1.0 expression. The XPath expression is evaluated against the S:Envelope element node to determine a true or false result. The expression SHOULD use the functions defined in Appendix I (where appropriate).
http://schemas.xmlsoap.org/2002/12/wsse#part	The contents of the <code><wsp:MessagePredicate></code> element is a list of required message parts that are identified using the set of pre-defined functions defined in Appendix II . The list of messages parts is evaluated against the S:Envelope element node to determine a true or false result; the expression is true if all the specified functions are non-empty.

/MessagePredicate/@{any}

Additional attributes MAY be specified but MUST NOT contradict the semantics of the owner element; if an attribute is not recognized, it SHOULD be ignored.

The following example illustrates the use of this declaration. In this example, the messages to which this assertion applies must contain exactly one WS-Security [[WS-Security](#)] <Security> header element.

```
<wsp:MessagePredicate wsp:Usage="wsp:Required">  
  count(wsp:GetHeader(./wsse:Security) = 1  
</wsp:MessagePredicate>
```

4. Security Considerations

It is strongly RECOMMENDED that policies and assertions be signed to prevent tampering.

It is RECOMMENDED that policies SHOULD NOT be accepted unless they are signed and have an associated security token to specify the signer has proper claims for the given policy. That is, a relying party shouldn't rely on a policy unless the policy is signed and presented with sufficient claims to pass the relying parties acceptance criteria.

It should be noted that the mechanisms described in this document could be secured as part of a SOAP message using WS-Security [[WS-Security](#)] or embedded within other objects using object-specific security mechanisms.

5. Acknowledgements

We would like to thank the following people for their contributions towards this specification:

- Debasish Banerjee, IBM
- Erik Christensen, Microsoft
- Francisco Curbera, IBM
- Giovanni Della-Libera, Microsoft
- Scott Konersmann, Microsoft
- Frank Leymann, IBM
- Steve Lucco, Microsoft
- Hiroshi Maruyama, IBM
- Steve Millet, Microsoft
- Henrik Frystyk Nielsen, Microsoft
- Keith Stobie, Microsoft
- Tony Storey, IBM
- Kent Tamura, IBM
- Sanjiva Weerawarana, IBM

6. References

[RFC 2119]

"Key words for use in RFCs to Indicate Requirement Levels," [RFC 2119](#), S. Bradner (editor), March 1997.

[RFC 2396]

T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

[SOAP11]

["SOAP: Simple Object Access Protocol 1.1,"](#) W3C Note, Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer, 08 May 2000.

[SOAP12]

["SOAP Version 1.2 Part 1: Messaging Framework,"](#) W3C Candidate Recommendation, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen (editors), 19 December 2002.

[WSDL]

["Web Services Description Language,"](#) IBM/Microsoft, 15 March 2001.

[WS-Policy]

["Web Services Policy Framework \(WS-Policy\),"](#) Don Box, Francisco Curbera, Maryann Hondo (Editor), Chris Kaler (Editor), Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Mark Nottingham, Claus von Riegen, and John Shewchuk, March 2003.

[WS-Security]

["Web Services Security \(WS-Security\),"](#) Bob Atkinson, Giovanni Della-Libera, Satoshi Hada, Maryann Hondo, Phillip Hallam-Baker, Chris Kaler (Editor), Johannes Klein, Brian LaMacchia, Paul Leach, John Manferdelli, Hiroshi Maruyama, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, John Shewchuk, and Dan Simon, 5 April 2002.

[XML]

["Extensible Markup Language \(XML\) 1.0 \(Second Edition\),"](#) W3C Recommendation, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler (editors), 6 October 2000.

[XML-NS]

["Namespaces in XML,"](#) W3C Recommendation, Tim Bray, Dave Hollander, and Andrew Layman (editors), 14 January 1999.

[XPATH]

["XML Path Language \(XPath\) Version 1.0,"](#) W3C Recommendation, James Clark and Steve DeRose (editors), 16 November 1999.

[XMLSchema1]

["XML Schema Part 1: Structures,"](#) W3C Recommendation, Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn (editors), 2 May 2001.

[XMLSchema2]

["XML Schema Part 2: Datatypes,"](#) W3C Recommendation, Paul V. Biron and Ashok Malhotra (editors), 2 May 2001.

Appendix I. XPath Expressions for Policies

This section defines a set of normative functions for use with XPath [XPATH] to simplify the path expressions that may be used within assertions to reference message elements.

We define the *evaluation context* for the XPath expressions to be the document containing the SOAP envelope. The namespaces in use at the point of the assertion element containing the XPath expression automatically defined for the context of the XPath expression. We do not define any variables, but we do define the functions (which work across both SOAP 1.1 [SOAP11] and SOAP 1.2 [SOAP12]) described below.

wsp: GetBody (node)	This function returns the SOAP envelope
---------------------	---

	<S:Body> element from the specified Envelope element.
wsp:IsInBody (node)	This function returns true if the specified node is in the SOAP envelope <S:Body> element from the specified Envelope element.
wsp:GetHeader (node)	This function returns the SOAP envelop <S:Header> element from the specified Envelope element.
wsp:IsInHeader (node)	This function returns true if the specified node is in the SOAP envelope <S:Header> element from the specified Envelope element.
wsp:RoleURIForHeaderBlock (node)	This function returns the SOAP role for the specified header block.
wsp:IsMandatoryHeaderBlock (node)	This function returns true/false depending on whether or not the specified header block is mandatory (<i>mustUnderstand</i> = true).
wsp:IsRoleURIForNext (node, string)	This function returns true/false depending on whether or not the specified role maps to the predefined "next" role for the version of SOAP used by the supplied message.
wsp:IsRoleURIForUltimateReceiver (node, string)	This function returns true/false depending on whether or not the specified role maps to the predefined "ultimate receiver" for the version of SOAP used by the supplied message.
wsp:GetNodesetForNode (node)	This function returns an XPath Node set for the node (this include the node, its attributes, all of its descendents and their attributes).

Appendix II. Message Part Selection Functions

This section defines a set of functions that are used with the "http://schemas.xmlsoap.org/2002/12/wsse#part" dialect for selecting parts of a message:

wsp:Body()	This function identifies the "body" of the message.
wsp:Header(x)	This function identifies a "header" whose name is the specified QName.