

Web Services Metadata Exchange (WS-MetadataExchange)

September 2004

Authors

Keith Ballinger, Microsoft
Don Box, Microsoft
Francisco Curbera (Editor), IBM
Srinivas Davanum, Computer Associates
Don Ferguson, IBM
Steve Graham, IBM
Canyang Kevin Liu, SAP
Frank Leymann, IBM
Brad Lovering, Microsoft
Anthony Nadalin, IBM
Mark Nottingham, BEA Systems
David Orchard, BEA Systems
Claus von Riegen, SAP
Jeffrey Schlimmer (Editor), Microsoft
Igor Sedukhin, Computer Associates
John Shewchuk, Microsoft
Bill Smith, Sun Microsystems, Inc.
Greg Truty, IBM
Sanjiva Weerawarana, IBM
Prasad Yendluri, webMethods

Copyright Notice

(c) 2004 [BEA Systems Inc.](#), [Computer Associates International, Inc.](#), [International Business Machines Corporation](#), [Microsoft Corporation, Inc.](#), [SAP AG](#), [Sun Microsystems](#), and [webMethods](#). All rights reserved.

Permission to copy and display the WS-MetadataExchange Specification (the "Specification"), in any medium without fee or royalty is hereby granted, provided that you include the following on ALL copies of the Specification that you make:

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

BEA Systems, Computer Associates, IBM, Microsoft, SAP, Sun, and webMethods (collectively, the "Authors") each agree to grant you a license, under royalty-free and otherwise reasonable, non-discriminatory terms and conditions, to their respective essential patent claims that they deem necessary to implement the WS-MetadataExchange Specification.

THE 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 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 SPECIFICATIONS.

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

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

Abstract

This specification defines messages to retrieve metadata associated with an endpoint.

Composable Architecture

By using the XML, SOAP [[SOAP 1.1](#), [SOAP 1.2](#)], and WSDL [[WSDL 1.1](#)] extensibility models, the Web services specifications (WS-*) are designed to be composed with each other to provide a rich set of tools to provide security in the Web services environment. This specification specifically relies on other Web services specifications to provide secure, reliable, and/or transacted message delivery and to express Web service and client policy.

Status

This specification is an initial public draft release and is provided for review and evaluation only. The authors hope to solicit your contributions and suggestions in the near future. The authors make no warranties or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction

- 1.1 Requirements
- 1.2 Example

2. Notations and Terminology

- 2.1 Notational Conventions
- 2.2 XML Namespaces
- 2.3 Compliance

3. Retrieving Metadata

- 3.1 Get Metadata
- 3.2 Get

4. Normative Protocol Binding

5. Security Considerations

6. Acknowledgements

7. References

Appendix I – Dialect URI Definitions

Appendix II – XML Schema

Appendix III – WSDL

1. Introduction

Web services use metadata to describe what other endpoints need to know to interact with them. Specifically, WS-Policy [[WS-Policy](#)] describes the capabilities, requirements, and general characteristics of Web services; WSDL [[WSDL 1.1](#)] describes abstract message operations, concrete network protocols, and endpoint addresses used by Web services; XML Schema [[XML Schema Part 1](#), [Part 2](#)] describes the structure and contents of XML-based messages received by and sent by Web services.

To bootstrap communication with Web services and to retrieve these and other types of metadata, this specification defines two request-response interactions. When the type of metadata sought is clearly known, e.g., WS-Policy, a requester may indicate that only that type should be returned; where additional types of metadata are being used, or are expected, or when a requester needs to retrieve all of the metadata relevant to subsequent interactions with an endpoint, a requester may indicate that all available metadata, regardless of their types, are expected.

The interactions defined herein are intended for the retrieval of metadata (i.e., service description information) only. They are not intended to provide a general purpose query or retrieval mechanism for other types of data associated with a service, such as state data, properties and attribute values, etc.

1.1 Requirements

This specification intends to meet the following requirements:

- Define a bootstrap mechanism for metadata-driven [[XML Schema](#), [WSDL](#), and [WS-Policy](#)] message exchange.
- Support future versions of known metadata formats.
- Allow new metadata formats to be added.
- Leverage other Web service specifications for secure, reliable, transacted message delivery.
- Support both SOAP 1.1 [[SOAP 1.1](#)] and SOAP 1.2 [[SOAP 1.2](#)] Envelopes.
- Enable description in WSDL 1.1 [[WSDL 1.1](#)].

1.2 Example

Table 1 illustrates a sample Get Metadata request for WS-Policy.

Table 1: Sample Get Metadata request message

```
(01) <s12:Envelope
(02)     xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04)     xmlns:wsx='http://schemas.xmlsoap.org/ws/2004/09/mex' >
```

```

(05) <s12:Header>
(06)   <wsa:Action>
(07)     http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request
(08)   </wsa:Action>
(09)   <wsa:MessageID>
(10)     uuid:73d7edfc-5c3c-49b9-ba46-2480caee43e9
(11)   </wsa:MessageID>
(12)   <wsa:ReplyTo>
(13)     <wsa:Address>http://client.example.com/MyEndpoint</wsa:Address>
(14)   </wsa:ReplyTo>
(15)   <wsa:To>http://server.example.org/YourEndpoint</wsa:To>
(16)   <ex:MyRefProp xmlns:ex='http://server.example.org/refs' >
(17)     78f2dc229597b529b81c4bef76453c96
(18)   </ex:MyRefProp>
(19) </s12:Header>
(20) <s12:Body>
(21)   <wsx:GetMetadata>
(22)     <wsx:Dialect>
(23)       http://schemas.xmlsoap.org/ws/2004/09/policy
(24)     </wsx:Dialect>
(25)   </wsx:GetMetadata>
(26) </s12:Body>
(27) </s12:Envelope>

```

Lines (06-08) in Table 1 indicate this is a Get Metadata request. Note that Lines (15-18) illustrate a typical pattern where the endpoint is identified by a wsa:To [\[WS-Addressing\]](#) header block (Line 15) as well as an application-specific header block (Lines 16-18). As Lines (22-24) indicate, this request is for the policy of the receiver; alternatively, it could include an identifier to request policy within a given target namespace.

Table 2 lists a sample response to the request in Table 1.

Table 2: Sample Get Metadata response message

```

(01) <s12:Envelope
(02)   xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)   xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04)   xmlns:wsp='http://schemas.xmlsoap.org/ws/2004/09/policy'
(05)   xmlns:wsx='http://schemas.xmlsoap.org/ws/2004/09/mex' >
(06) <s12:Header>
(07)   <wsa:Action>

```

```

(08)      http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
(09)      </wsa:Action>
(10)      <wsa:RelatesTo>
(11)          uuid:73d7edfc-5c3c-49b9-ba46-2480caee43e9
(12)      </wsa:RelatesTo>
(13)      <wsa:To>http://client.example.com/MyEndpoint</wsa:To>
(14) </s12:Header>
(15) <s12:Body>
(16)   <wsx:Metadata>
(17)     <wsx:MetadataSection
(18)       Dialect='http://schemas.xmlsoap.org/ws/2004/09/policy' >
(19)       <wsp:Policy
(20)         xmlns:wsse='http://schemas.xmlsoap.org/ws/2002/12/secext' >
(21)         <wsp:ExactlyOne>
(22)           <wsse:SecurityToken>
(23)             <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
(24)           </wsse:SecurityToken>
(25)           <wsse:SecurityToken>
(26)             <wsse:TokenType>wsse:X509v3</wsse:TokenType>
(27)           </wsse:SecurityToken>
(28)         </wsp:ExactlyOne>
(29)       </wsp:Policy>
(30)     </wsx:MetadataSection>
(31)   </wsx:Metadata>
(32) </s12:Body>
(33) </s12:Envelope>

```

Lines (07-09) in Table 2 indicate this message is a response to a Get Metadata request, and Lines (10-12) indicate that it is a response to the request in Table 1. Lines (16-31) contain a single metadata section (Lines 17-30); Line (18) indicates the metadata in this section is of type, or dialect, WS-Policy. Lines (19-29) contain the policy corresponding to the receiver of the Get Metadata request in Table 1.

The corresponding WSDL [[WSDL 1.1](#)] for use with this or other bindings is below. (Refer to Appendix III – WSDL for the complete definition.)

```

<wsdl:message name='GetMetadataMsg' >
  <wsdl:part name='body' element='tns:GetMetadata' />
</wsdl:message>
<wsdl:message name='GetMetadataResponseMsg' >

```

```

<wsdl:part name='body' element='tns:Metadata' />
</wsdl:message>

<wsdl:portType name='MetadataExchange' >
  <wsdl:operation name='GetMetadata' >
    <wsdl:input
      message='tns:GetMetadataMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request' />
    <wsdl:output
      message='tns:GetMetadataResponseMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response' />
    </wsdl:operation>
    ...
  </wsdl:portType>

```

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](#)].

This specification uses the following syntax to define normative outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- Characters are appended to elements and attributes to indicate cardinality:
 - "?" (0 or 1)
 - "*" (0 or more)
 - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "[" and "]" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. If an extension is not recognized it SHOULD be ignored.
- XML namespace prefixes (see Table 3) are used to indicate the namespace of the element being defined.

2.2 XML Namespaces

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

```
http://schemas.xmlsoap.org/ws/2004/09/mex
```

Table 3 lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 3: Prefixes and XML namespaces used in this specification

Prefix	XML Namespace	Specification(s)
s	(Either SOAP 1.1 or 1.2)	(Either SOAP 1.1 or 1.2)
s11	http://schemas.xmlsoap.org/soap/envelope/	SOAP 1.1 [SOAP 1.1]
s12	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2 [SOAP 1.2]
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	WS-Addressing [WS-Addressing]
wSDL	http://schemas.xmlsoap.org/wSDL/	WSDL [WSDL 1.1]
wsse	http://schemas.xmlsoap.org/ws/2002/12/secext	WS-SecurityPolicy [WS-SecurityPolicy]
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	WS-Policy [WS-Policy]
wsx	http://schemas.xmlsoap.org/ws/2004/09/mex	This specification
xs	http://www.w3.org/2001/XMLSchema	XML Schema [Part 1 , 2]

2.3 Compliance

An implementation is not compliant with this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined herein. A SOAP Node MUST NOT use the XML namespace identifier for this specification (listed in [Section 2.2](#)) within SOAP Envelopes unless it is compliant with this specification.

Normative text within this specification takes precedence over normative outlines, which in turn take precedence over the XML Schema [[XML Schema Part 1](#), [Part 2](#)] and WSDL [[WSDL 1.1](#)] descriptions.

3. Retrieving Metadata

3.1 Get Metadata

To retrieve a service's metadata, a requester MAY send a Get Metadata request message to an endpoint. The normative outline for a Get Metadata request is:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
```

```

    <wsa:To>xs:anyURI</wsa:To>
    ...
</s:Header>
<s:Body>
  <wsx:GetMetadata ...>
    [<wsx:Dialect>xs:anyURI</wsx:Dialect>
      [<wsx:Identifier>xs:anyURI</wsx:Identifier>]?
    ]?
  </wsx:GetMetadata>
</s:Body>
</s:Envelope>

```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required header block MUST contain the value indicated. If in addition to this message header a SOAP Action URI is used in the binding for SOAP, the value indicated MUST be used for that URI.

/s:Envelope/s:Body/wsx:GetMetadata/wsx:Dialect

When this element is present, the response MUST include only Metadata Sections with the indicated dialect; if the receiver does not have any Metadata Sections of the indicated dialect, the response MUST include zero Metadata Sections. When this element is not present, the implied value is any dialect.

/s:Envelope/s:Body/wsx:GetMetadata/wsx:Identifier

When this element is present, the response MUST include only Metadata Sections with the indicated identifier; if the receiver does not have any Metadata Sections of the indicated identifier, the response MUST include zero Metadata Sections. When this element is not present, the implied value is any identifier. This element MUST NOT be present unless ./wsx:Dialect is present.

Other message information headers defined by WS-Addressing [[WS-Addressing](#)] MAY be included in the request and response messages, according to the usage and semantics defined in WS-Addressing.

An endpoint MAY respond with a fault message using the standard fault codes defined in WS-Addressing [[WS-Addressing](#)] (e.g., wsa:ActionNotSupported).

If an endpoint accepts a Get Metadata request, it MUST reply with a Get Metadata response message. The normative outline for a Get Metadata response is:

```

<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
    </wsa:Action>
    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
    <wsa:To>xs:anyURI</wsa:To>

```

```

...
</s:Header>
<s:Body ...>
  <wsx:Metadata ...>
    [ <wsx:MetadataSection Dialect='xs:anyURI'
      [ Identifier='xs:anyURI' ]?
      ...>
      [
        <dialectSpecificElementName>...</dialectSpecificElementName>+
        |
        <wsx:MetadataReference ...>
          endpoint-reference
        </wsx:MetadataReference>
        |
        <wsx:Location>xs:anyURI</wsx:Location>
      ]
    </wsx:MetadataSection>]*
  </wsx:Metadata>
</s:Body>
</s:Envelope>

```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required header block **MUST** contain the value indicated. If in addition to this message header a SOAP Action URI is used in the binding for SOAP, the value indicated **MUST** be used for that URI.

/s:Envelope/s:Body/wsx:Metadata

This required element contains zero or more `wsx:MetadataSection` elements, one per distinct unit of metadata that describes some aspect of the target service.

Because a Get Metadata request does not necessarily qualify the types or quantities of metadata, a Get Metadata response could potentially include a large amount of information. To ensure scalability, this specification defines two alternative mechanisms for granting access to the available metadata: directly copying the information into the response message, and providing a reference to an endpoint (Metadata Reference) or URL (Metadata Location) where the information can be retrieved. These mechanisms are described in detail below. It is **RECOMMENDED** that when the amount of metadata to be returned is large, the response message return Metadata References or Locations instead of the actual information.

/s:Envelope/s:Body/wsx:Metadata/wsx:MetadataSection

This repeating element represents a single unit of metadata (e.g., a WSDL definitions element, an XSD schema element) for a given scope/domain/namespace. This element contains exactly one child element, either the embedded XML for the

metadata, an endpoint reference that can be dereferenced using a Get operation to fetch the metadata (See 3.2 Get), or a URL that can be dereferenced using an HTTP GET.

/s:Envelope/s:Body/wsx:Metadata/wsx:MetadataSection/@Dialect

This required attribute (of type xs:anyURI) specifies the format and version of this metadata section (e.g., WSDL version 1.1). This specification defines several values for this attribute in Appendix I – Dialect URI Definitions.

/s:Envelope/s:Body/wsx:Metadata/wsx:MetadataSection/@Identifier

This optional attribute (of type xs:anyURI) specifies the identifier for the scope/domain/namespace of this metadata section. Its interpretation is dialect-specific, but @Identifier is typically the name of this instance of metadata whereas @Dialect is the type of the metadata.

- For XML Schema, this is the value of the xs:schema/@targetNamespace value.
- For WSDL, this is the value of the wsdl:definitions/@targetNamespace attribute. When absent, the WSDL definitions element MUST contain exactly one wsdl:service element that describes the target service.
- For WS-Policy expressions, this is either the value of the wsp:Policy/@TargetNamespace or the URI of the policy expression as used in the wsp:PolicyReference element.

If there is more than one Metadata Section with the requested Identifier, e.g., more than one XML Schema in the same target namespace, returning them all, one per metadata section, is explicitly encouraged.

/s:Envelope/s:Body/wsx:Metadata/wsx:MetadataSection/wsx:MetadataReference

When this element is present, it contains an endpoint reference that may be used with a Get request to fetch the metadata for the section's dialect and identifier. (See 3.2 Get.) When this element is present, it MUST have no element siblings.

/s:Envelope/s:Body/wsx:Metadata/wsx:MetadataSection/wsx:Location

When this element is present, it contains a URL, and the metadata MUST be retrievable from that URL using the primary access mechanism for the scheme of the URL (e.g., for an HTTP URL, the metadata MUST be retrievable by sending an HTTP GET request to the URL). When this element is present, it MUST have no element siblings.

/s:Envelope/s:Body/wsx:Metadata/wsx:MetadataSection [not(wsx:MetadataReference | wsx:Location)]

When any element other than wsx:MetadataReference or wsx:Location is present, the element is to be interpreted as the representation of the metadata associated with the section's dialect and identifier.

Table 4 lists a sample Get Metadata request for all types of metadata.

Table 4: Sample Get Metadata request message

```
(01) <s12:Envelope
(02)     xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04)     xmlns:wsx='http://schemas.xmlsoap.org/ws/2004/09/mex' >
(05) <s12:Header>
(06)     <wsa:Action>
```

```

(07) http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request
(08)   </wsa:Action>
(09)   <wsa:MessageID>
(10)     uuid:a6e37bfb-f324-4e71-b33a-4f6d5c6027f4
(11)   </wsa:MessageID>
(12)   <wsa:ReplyTo>
(13)     <wsa:Address>http://client.example.com/MyEndpoint</wsa:Address>
(14)   </wsa:ReplyTo>
(15)   <wsa:To>http://server.example.org/YourEndpoint</wsa:To>
(16) </s12:Header>
(17) <s12:Body>
(18)   <wsx:GetMetadata />
(19) </s12:Body>
(20) </s12:Envelope>

```

Lines (06-08) in Table 4 indicate this is a Get Metadata request. Because no dialect or identifier are specified in the body (Line 18), this is a request for all metadata for the receiver.

Table 5 lists a sample response to the request in Table 4.

Table 5: Sample Get Metadata response message

```

(01) <s12:Envelope
(02)   xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)   xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
(04)   xmlns:wsx='http://schemas.xmlsoap.org/ws/2004/09/mex' >
(05) <s12:Header>
(06)   <wsa:Action>
(07) http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
(08)   </wsa:Action>
(09)   <wsa:RelatesTo>
(10)     uuid:a6e37bfb-f324-4e71-b33a-4f6d5c6027f4
(11)   </wsa:RelatesTo>
(12)   <wsa:To>http://client.example.com/MyEndpoint</wsa:To>
(13) </s12:Header>
(14) <s12:Body>
(15)   <wsx:Metadata>
(16)     <wsx:MetadataSection Dialect='http://schemas.xmlsoap.org/wsdl/'>
(17)       <wsdl:definitions
(18)         name='StockQuote'

```

```

(19)         targetNamespace='http://server.example.org/stockquote '
(20)         xmlns:tns='http://server.example.org/stockquote '
(21)         xmlns:wSDL='http://schemas.xmlsoap.org/wSDL/'
(22)         xmlns:wsdl='http://schemas.xmlsoap.org/wSDL/soap/' >
(23)     <wSDL:import namespace='http://server.example.org/stockquote '
(24)                 location='http://server.example.org/stockquote' />
(25)     <wSDL:portType name='StockQuotePortType'>
(26)         <wSDL:operation name='GetLastTradePrice'>
(27)             <wSDL:input message='tns:GetLastTradePriceInput' />
(28)             <wSDL:output message='tns:GetLastTradePriceOutput' />
(29)         </wSDL:operation>
(30)     </wSDL:portType>
(31)
(32)     <wSDL:service name='StockQuoteService'>
(33)         <wSDL:port name='StockQuotePort'
(34)                 binding='tns:StockQuoteBinding' >
(35)             <wsdl:address
(36)                 location='http://server.example.org/YourEndpoint' />
(37)             </wSDL:port>
(38)         </wSDL:service>
(39) </wSDL:definitions>
(40) </wsx:MetadataSection>
(41) <wsx:MetadataSection
(42)     Dialect='http://www.w3.org/2001/XMLSchema'
(43)     Identifier='urn:fabrikam123:schemas:sq'>
(44)     <wsx:MetadataReference>
(45)         <wsa:Address>
(46)             http://www.fabrikam123.com/schemas/sq
(47)         </wsa:Address>
(48)     </wsx:MetadataReference>
(49) </wsx:MetadataSection>
(50) </wsx:Metadata>
(51) </s12:Body>
(52) </s12:Envelope>

```

Lines (06-08) in Table 5 indicate this message is a response to a Get Metadata request, and Lines (09-11) indicate that it is a response to the request identified by the message

identifier in Table 4. Lines (15-50) is the `wxs:Metadata` element that contains two metadata sections. The first section (Lines 16-40) contains a WSDL document that contains one `wSDL:service` element that describes the target service. Lines (41-49) describe a second metadata section for an XML Schema that is sent by reference using the `wsa:MetadataReference` element. The receiver may fetch the metadata by issuing a Get operation against the specified endpoint. Note that this example includes only a single WSDL; if there is more than one WSDL in the requested target namespace, returning all of them, one per metadata section, is explicitly encouraged.

3.2 Get

To retrieve a referenced Metadata Section, a requester MAY send a Get request message to a Metadata Reference. Get fetches a one-time snapshot of the metadata, according to the metadata type (`@Dialect`) and identifier specified in the Metadata Section. To facilitate intelligent intermediaries, all implementations of the Get operation must be "safe", as defined in RFC 2616 [RFC 2616]. Specifically, safe operations are required to have no significant semantic side-effects on the service, including such actions as the acquisition of long-lived locks.

The normative outline for a Get request is:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Request
    </wsa:Action>
    <wsa:MessageID>xs:anyURI</wsa:MessageID>
    <wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body/>
</s:Envelope>
```

The following describes additional, normative constraints on the outline listed above:

`/s:Envelope/s:Header/wsa:Action`

This required element MUST contain the value indicated. If in addition to this message header a SOAP Action URI is used in the binding for SOAP, the value indicated MUST be used for that URI.

`/s:Envelope/s:Body`

The `s:Body` element for this message has no children.

Other message information headers defined by WS-Addressing [WS-Addressing] MAY be included in the request and response messages, according to the usage and semantics defined in WS-Addressing.

An endpoint MAY respond with a fault message using the standard fault codes defined in WS-Addressing [WS-Addressing] (e.g., `wsa:ActionNotSupported`).

If the endpoint accepts a Get request, it MUST reply with a Get response message. The normative outline for a Get response is:

```
<s:Envelope ...>
  <s:Header ...>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Response
    </wsa:Action>
    <wsa:RelatesTo>xs:anyURI</wsa:RelatesTo>
    <wsa:To>xs:anyURI</wsa:To>
    ...
  </s:Header>
  <s:Body> ... </s:Body>
</s:Envelope>
```

The following describes additional, normative constraints on the outline listed above:

/s:Envelope/s:Header/wsa:Action

This required header block MUST contain the value indicated. If in addition to this message header a SOAP Action URI is used in the binding for SOAP, the value indicated MUST be used for that URI.

/s:Envelope/s:Body/child::*[position()=1]

The metadata itself MUST be the initial child element of the s:Body element of the response message.

Other message information headers defined by WS-Addressing [[WS-Addressing](#)] MAY be included in the request and response messages, according to the usage and semantics defined in WS-Addressing.

The corresponding WSDL [[WSDL 1.1](#)] definition of the Get operation is below. (Refer to Appendix III – WSDL for the complete definition.)

```
<wsdl:message name='GetMsg' />
<wsdl:message name='GetResponseMsg' >
  <wsdl:part name='body' type='tns:anyXml' />
</wsdl:message>

<wsdl:portType name='MetadataExchange' >
  ...
  <wsdl:operation name='Get' >
    <wsdl:input
      message='tns:GetMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Request' />
    <wsdl:output
      message='tns:GetResponseMsg'
```

```

        wsa:Action=
            'http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Response' />
    </wsdl:operation>
</wsdl:portType>

```

Table 6 lists a sample Get request.

Table 6: Sample Get request message

```

(01) <s12:Envelope
(02)     xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing' >
(04)   <s12:Header>
(05)     <wsa:Action>
(06)       http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Request
(07)     </wsa:Action>
(08)     <wsa:MessageID>
(09)       uuid:3e3aac89-ba01-4568-80bf-273c2bc14d1c
(10)     </wsa:MessageID>
(11)     <wsa:ReplyTo>
(12)       <wsa:Address>http://client.example.com/MyEndpoint</wsa:Address>
(13)     </wsa:ReplyTo>
(14)     <wsa:To>http://www.fabrikam123.com/schemas/sq</wsa:To>
(15)   </s12:Header>
(16)   <s12:Body/>
(17) </s12:Envelope>

```

Lines (05-07) in Table 6 indicate this is a Get request message. Line (14) indicates this is a copy of the metadata included by reference in Lines (41-49) in Table 5.

Table 7 lists a sample response to the request in Table 6.

Table 7: Sample Get response message

```

(01) <s12:Envelope
(02)     xmlns:s12='http://www.w3.org/2003/05/soap-envelope'
(03)     xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing' >
(04)   <s12:Header>
(05)     <wsa:Action>
(06)       http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Response
(07)     </wsa:Action>
(08)     <wsa:RelatesTo>
(09)       uuid:3e3aac89-ba01-4568-80bf-273c2bc14d1c
(10)     </wsa:RelatesTo>

```

```

(11)    <wsa:To>http://client.example.com/MyEndpoint</wsa:To>
(12)    </s12:Header>
(13)    <s12:Body>
(14)        <xs:schema targetNamespace='urn:fabrikam123:schemas:sq'
(15)            xmlns:xs='http://www.w3.org/2001/XMLSchema'
(16)            elementFormDefault='qualified'
(17)            blockDefault='#all' >
(18)        <xs:element name='StockQuote' >
(19)            <xs:complexType>
(20)                <xs:sequence>
(21)                    <xs:element name='LastTrade' type='xs:decimal' />
(22)                    <xs:element name='TradeTime' type='xs:dateTime' />
(23)                    <xs:element name='Change' type='xs:float' />
(24)                    <xs:element name='PrevClose' type='xs:decimal' />
(25)                    <xs:element name='Open' type='xs:decimal' />
(26)                </xs:sequence>
(27)            </xs:complexType>
(28)        </xs:element>
(29)    </xs:schema>
(30) </s12:Body>
(31) </s12:Envelope>

```

Lines (05-07) in Table 7 indicate this message is a response to a Get request, and Lines (08-10) indicate that it is a response to the request in Table 6. Lines (14-29) contain the requested XML Schema. Note that the target namespace in Line (14) matches the Identity of the Metadata Section listed in Line (43) in Table 5.

4. Normative Protocol Binding

A binding for the messages described herein to SOAP 1.1 [[SOAP 1.1](#)] over HTTP as constrained by the Basic Profile 1.0 [[BP 1.0](#)] is RECOMMENDED as a means to bootstrap communication. A Web service is free to support these messages over other bindings in addition to, or in place of, this binding as specified by WSDL [[WSDL 1.1](#)], policies, or other mechanisms. In the absence of an explicit specification stating that a different binding must be used, the default SOAP 1.1 over HTTP binding defined here is assumed to apply.

5. Security Considerations

It is strongly RECOMMENDED that the communication between Web services be secured using the mechanisms described in WS-Security [[WS-Security](#)]. In order to properly secure messages, the body and all relevant headers need to be included in the signature. Specifically, any standard messaging headers, such as those from WS-Addressing [[WS-Addressing](#)], need to be signed with the body in order to "bind" the two together.

Different security mechanisms may be desired depending on the frequency of messages. For example, for infrequent messages, public key technologies may be adequate for integrity and confidentiality. However, for high-frequency events, it may be more performant to establish a security context for the events using the mechanisms described in WS-Trust [[WS-Trust](#)] and WS-SecureConversation [[WS-SecureConversation](#)]. It should be noted that if a shared secret is used it is RECOMMENDED that derived keys be used to strengthen the secret as described in WS-SecureConversation.

Requests for metadata that are not available to anonymous parties are strongly RECOMMENDED to require usage of WS-Security so that the requester can be authenticated and authorized to access the indicated metadata. Similarly, integrity and confidentiality SHOULD be used whenever metadata has restricted access.

Recipients of metadata are RECOMMENDED to validate the signature to authenticate and verify the integrity of the data. Specifically, recipients SHOULD verify that the sender has the right to "speak" for the metadata. This is important because some metadata, such as schemas, have embedded target URIs that might be outside the scope of the sender.

Additionally, some metadata formats, such as policies [[WS-Policy](#)], may have embedded security semantics. These SHOULD be verified using the same considerations outlined in this section.

The following list summarizes common classes of attacks that apply to this protocol and identifies the mechanism to prevent/mitigate the attacks:

- **Message alteration** – Alteration is prevented by including signatures of the message information using WS-Security.
- **Message disclosure** – Confidentiality is preserved by encrypting sensitive data using WS-Security.
- **Key integrity** – Key integrity is maintained by using the strongest algorithms possible (by comparing secured policies – see WS-Policy and WS-SecurityPolicy [[WS-SecurityPolicy](#)]).
- **Authentication** – Authentication is established using the mechanisms described in WS-Security and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- **Accountability** – Accountability is a function of the type of and strength of the key and algorithms being used. In many cases, a strong symmetric key provides sufficient accountability. However, in some environments, strong PKI signatures are required.
- **Availability** – Metadata services are subject to a variety of availability attacks such as application-level denial of service. It is recommended that the mechanisms described in WS-Security be considered as mitigations for some forms of attacks. Other attacks, such as network-level denial of service are harder to avoid. Note that both of these classes of attack are outside the scope of this specification.
- **Replay** – Messages may be replayed for a variety of reasons. To detect and eliminate this attack, mechanisms should be used to identify replayed messages such as the timestamp/nonce outlined in WS-Security. Alternatively, and optionally, other technologies, such as sequencing, can also be used to prevent replay of application messages.

6. Acknowledgements

This specification has been developed as a result of joint work with many individuals and teams, including: Luis Felipe Cabrera (Microsoft), Erik Christensen (Microsoft), Timm Falter (SAP), Jeffrey Frey (IBM), Omri Gazitt (Microsoft), Alan Geller (Microsoft), Martin Gudgin (Microsoft), Maryann Hondo (IBM), Chris Kaler (Microsoft), Dave Langworthy (Microsoft), Andrew Layman (Microsoft), Steve Millet (Microsoft), Henrik Frystyk Nielsen (Microsoft), Sanjay Patil (SAP), Vladimir Savchenko (SAP), Ramesh Seshadri (Microsoft), Chris Sharp (IBM), Eugene Sindambiwe (SAP), Tony Storey (IBM), Marvin Theimer (Microsoft), Sara Wong (Microsoft), Alexander Zubev (SAP).

7. References

[BP 1.0]

K. Ballinger, et al, "[Basic Profile Version 1.0](#)," April 2004.

[RFC 2119]

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

[RFC 2616]

R. Fielding, et al, "Hypertext Transfer Protocol -- HTTP/1.1," [RFC 2616](#), UC Irvine, June 1999.

[SOAP 1.1]

D. Box, et al, "[Simple Object Access Protocol \(SOAP\) 1.1](#)," May 2000.

[SOAP 1.2]

M. Gudgin, et al, "[SOAP Version 1.2 Part 1: Messaging Framework](#)," June 2003.

[WS-Addressing]

D. Box, et al, "[Web Services Addressing \(WS-Addressing\)](#)," August 2004.

[WS-Policy]

S. Bajaj, et al, "[Web Services Policy Framework \(WS-Policy\)](#)," September 2004.

[WS-PolicyAttachment]

S. Bajaj, et al, "[Web Services Policy Attachment \(WS-PolicyAttachment\)](#)," September 2004.

[WS-SecureConversation]

S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," May 2004.

[WS-Security]

A. Nadalin, et al, "[Web Services Security: SOAP Message Security 1.0 \(WS-Security 2004\)](#)," March 2004.

[WS-SecurityPolicy]

G. Della-Libera, et al, "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)," December 2002.

[WS-Trust]

S. Anderson, et al, "[Web Services Trust Language \(WS-Trust\)](#)," May 2004.

[WSDL 1.1]

E. Christensen, et al, "[Web Services Description Language \(WSDL\) 1.1](#)," March 2001.

[XML Schema, Part 1]

H. Thompson, et al, "[XML Schema Part 1: Structures](#)," May 2001.

[XML Schema, Part 2]

P. Biron, et al, "[XML Schema Part 2: Datatypes](#)," May 2001.

Appendix I – Dialect URI Definitions

Table 8 defines several values for @Dialect; other specifications are expected to define values for @Dialect for other metadata formats and/or versions.

Table 8: Dialect URI Definitions

Dialect URI	Metadata Format
http://www.w3.org/2001/XMLSchema	XML Schema Version 1.0
http://schemas.xmlsoap.org/wsdl/	WSDL 1.1
http://schemas.xmlsoap.org/ws/2004/09/policy	WS-Policy expression
http://schemas.xmlsoap.org/ws/2004/09/policy/attachment	WS-PolicyAttachment wsp:PolicyAttachment element
http://schemas.xmlsoap.org/ws/2004/09/mex	This specification

The "<http://schemas.xmlsoap.org/ws/2004/09/mex>" dialect indicates a `wsx:Metadata` element is nested within a metadata section; when used in conjunction with `wsx:MetadataReference` or `wsx:Location`, this allows the service to indicate a source for additional metadata.

Appendix II – XML Schema

A normative copy of the XML Schema [[XML Schema Part 1](#), [Part 2](#)] description for this specification may be retrieved by resolving the XML namespace URI for this specification (listed in Section 2.2 XML Namespaces). A non-normative copy of the XML Schema description is listed below for convenience.

```
<xs:schema
  targetNamespace='http://schemas.xmlsoap.org/ws/2004/09/mex'
  xmlns:tns='http://schemas.xmlsoap.org/ws/2004/09/mex'
  xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  elementFormDefault='qualified'
  blockDefault='#all' >

  <xs:import
    namespace='http://schemas.xmlsoap.org/ws/2004/08/addressing'
    schemaLocation='http://schemas.xmlsoap.org/ws/2004/08/addressing/' />

  <!-- Get Metadata request -->
  <xs:element name='GetMetadata' >
    <xs:complexType>
      <xs:sequence>
```

```

    <xs:element ref='tns:Dialect' minOccurs='0' />
    <xs:element ref='tns:Identifier' minOccurs='0' />
  </xs:sequence>
  <xs:anyAttribute namespace='##other' processContents='lax' />
</xs:complexType>
</xs:element>

<xs:element name='Dialect' type='xs:anyURI' />
<xs:element name='Identifier' type='xs:anyURI' />

<!-- Get Metadata response -->
<xs:element name='Metadata' >
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='tns:MetadataSection'
        minOccurs='0'
        maxOccurs='unbounded' />
    </xs:sequence>
    <xs:anyAttribute namespace='##other' processContents='lax' />
  </xs:complexType>
</xs:element>

<xs:element name='MetadataSection' >
  <xs:complexType>
    <xs:choice>
      <xs:any namespace='##other'
        processContents='lax'
        minOccurs='0'
        maxOccurs='unbounded' />
      <xs:element ref='tns:MetadataReference' />
      <xs:element ref='tns:Location' />
    </xs:choice>
    <xs:attribute name='Dialect' type='xs:anyURI' use='required' />
    <xs:attribute name='Identifier' type='xs:anyURI' />
    <xs:anyAttribute namespace='##other' processContents='lax' />
  </xs:complexType>
</xs:element>

```

```

<xs:element name='MetadataReference'
            type='wsa:EndpointReferenceType' />

<xs:element name='Location'
            type='xs:anyURI' />

<!-- count(/s:Envelope/s:Body/*) = 0 for Get request -->

<!-- Get Response returns xs:any -->

<xs:complexType name='AnyXmlType' >
  <xs:sequence>
    <xs:any namespace='##any' processContents='lax' />
  </xs:sequence>
  <xs:anyAttribute namespace='##any' processContents='lax' />
</xs:complexType>

</xs:schema>

```

Appendix III – WSDL

A normative copy of the WSDL [\[WSDL 1.1\]](#) description for this specification can be retrieved from the following address:

```
http://schemas.xmlsoap.org/ws/2004/09/mex/metadataexchange.wsdl
```

A non-normative copy of the WSDL description is listed below for convenience.

```

<wsdl:definitions
  targetNamespace='http://schemas.xmlsoap.org/ws/2004/09/mex'
  xmlns:tns='http://schemas.xmlsoap.org/ws/2004/09/mex'
  xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'
  xmlns:wSDL='http://schemas.xmlsoap.org/wSDL/'
  xmlns:xs='http://www.w3.org/2001/XMLSchema' >

  <wsdl:types>
    <xs:schema
      targetNamespace='http://schemas.xmlsoap.org/ws/2004/09/mex' >
      <xs:include schemaLocation='metadataexchange.xsd' />
    </xs:schema>
  </wsdl:types>

```

```
<wsdl:message name='GetMetadataMsg' >
  <wsdl:part name='body' element='tns:GetMetadata' />
</wsdl:message>
<wsdl:message name='GetMetadataResponseMsg' >
  <wsdl:part name='body' element='tns:Metadata' />
</wsdl:message>

<wsdl:message name='GetMsg' />

<wsdl:message name='GetResponseMsg' >
  <wsdl:part name='body' type='tns:AnyXmlType' />
</wsdl:message>

<wsdl:portType name='MetadataExchange' >
  <wsdl:operation name='GetMetadata' >
    <wsdl:input
      message='tns:GetMetadataMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request' />
    <wsdl:output
      message='tns:GetMetadataResponseMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response' />
    </wsdl:operation>
  <wsdl:operation name='Get' >
    <wsdl:input
      message='tns:GetMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Request' />
    <wsdl:output
      message='tns:GetResponseMsg'
      wsa:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/Get/Response' />
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

