

# **Web Services Metadata Exchange (WS-MetadataExchange)**

## **Version 1.1**

**August 2006**

### **Authors**

Keith Ballinger, Microsoft  
Bobby Bissett, Sun Microsystems, Inc.  
Don Box, Microsoft  
Francisco Curbera (Editor), IBM  
Don Ferguson, IBM  
Steve Graham, IBM  
Canyang Kevin Liu, SAP  
Frank Leymann, IBM  
Brad Lovering, Microsoft  
Raymond McCollum, Microsoft  
Anthony Nadalin, IBM  
David Orchard, BEA Systems  
Savas Parastatidis (Editor), Microsoft  
Claus von Riegen, SAP  
Jeffrey Schlimmer (Editor), Microsoft  
John Shewchuk, Microsoft  
Bill Smith, Sun Microsystems, Inc.  
Greg Truty, IBM  
Asir Vedamuthu, Microsoft  
Sanjiva Weerawarana, IBM<sup>1</sup>  
Kirk Wilson, Computer Associates  
Prasad Yendluri, webMethods

### **Copyright Notice**

(c) 2004-2006 [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 Web Services Metadata Exchange Specification (the "Specification", which includes WSDL and schema documents), 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 one of the Authors' websites.
2. The copyright notice as shown in the Specification.

---

<sup>1</sup> Currently with WSO2.

[BEA Systems Inc.](#), [Computer Associates International, Inc.](#), [International Business Machines Corporation](#), [Microsoft Corporation, Inc.](#), [SAP AG](#), [Sun Microsystems](#), 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 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 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 Specification will at all times remain with the Authors.

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

## **Abstract**

This specification defines how metadata associated with a Web service endpoint can be represented as [[WS-Transfer](#)] resources, how metadata can be embedded in [[WS-Addressing 2004](#), [WS-Addressing 1.0 Core](#)] endpoint references, and how metadata could be retrieved from a Web service endpoint.

## **Composable Architecture**

The Web services specifications (WS-\*) are designed to be composed with each other to provide a rich set of tools for 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 metadata.

## **Status**

This specification is a 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. Notation**

- 2.1 XML Namespaces
- 2.2 Notational Conventions
- 2.3 Compliance

### **3. Metadata Resources**

## **4. Web Services Metadata**

### **5. Retrieving Metadata**

5.1 WS-Transfer Get

5.2 Get Metadata

### **6. Metadata in Endpoint References**

### **7. Bootstrapping Metadata Retrieval**

### **8. Security**

### **9. Acknowledgements**

### **10. References**

#### **Appendix I – XML Schema**

#### **Appendix II – 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 this specification defines how metadata can be treated as [[WS-Transfer](#)] resources for retrieval purposes, how metadata can be embedded in Web service endpoint references, and how Web service endpoints can optionally support a request-response interaction for the retrieval of metadata. 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 mechanisms defined herein are intended for the retrieval of metadata (i.e., Web 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 Web service, such as state data, properties and attribute values, etc.

### **1.1 Requirements**

This specification intends to meet the following requirements:

- Define an encapsulation format for metadata.
- Treat the metadata about a Web service endpoint as [[WS-Transfer](#)] resources.
- Define an optional 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](#)] of the optional request-response interaction.

## 1.2 Example

Table 1 illustrates a sample [[WS-Transfer](#)] Get request for a resource's representation.

**Table 1: Sample Get request message.**

```
(01) <s11:Envelope
(02)   xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
(03)   xmlns:wsa10='http://www.w3.org/2005/08/addressing'>
(04)   <s11:Header>
(05)     <wsa10:Action>
(06)       http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
(07)     </wsa10:Action>
(08)     <wsa10:To>http://services.example.org/stockquote/metadata</wsa10:To>
(09)     <wsa10:ReplyTo>
(10)       <wsa10:Address>http://client.example.org</wsa10:Address>
(11)     </wsa10:ReplyTo>
(12)     <wsa10:MessageID>
(13)       urn:uuid:1cecc121a-82fe-41da-87e1-3b23f254f128
(14)     </wsa10:MessageID>
(15)   </s11:Header>
(16)   <s11:Body />
(17) </s11:Envelope>
```

The sample request message of Table 1 is a [[WS-Transfer](#)] request for the retrieval of a resource's representation. In this case, the requested representation is the WS-Metadata Exchange Metadata element about a Web service endpoint. The fact that the resource's representation is a mex:Metadata element may be known to the requestor but is not explicitly encoded in the request message

Table 2 illustrates a sample response to the request of Table 1.

**Table 2: Sample response message with metadata.**

```
(01) <s11:Envelope
(02)   xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
(03)   xmlns:wsa10='http://www.w3.org/2005/08/addressing'
(04)   xmlns:mex='http://schemas.xmlsoap.org/ws/2004/09/mex'
(05)   xmlns: wsp='http://schemas.xmlsoap.org/ws/2004/09/policy'
(06)   xmlns: wsdl='http://schemas.xmlsoap.org/wsdl/'>
(07)   <s11:Header>
(08)     <wsa10:Action>
(09)       http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
(10)     </wsa10:Action>
(11)     <wsa10:To>http://client.example.org</wsa10:To>
(12)     <wsa10:RelatesTo>
(13)       urn:uuid:1cecc121a-82fe-41da-87e1-3b23f254f128
(14)     </wsa10:RelatesTo>
(15)   </s11:Header>
(16)   <s11:Body>
(17)     <mex:Metadata>
```

```

(18)      <mex:MetadataSection Dialect='http://schemas.xmlsoap.org/wsdl/'>
(19)          <wsdl:definitions
(20)              name='StockQuote'
(21)                  targetNamespace='http://services.example.org/stockquote'
(22)                  xmlns:tns='http://services.example.org/stockquote'
(23)                  xmlns:msgs='http://services.example.org/stockquote/schemas'
(24)                  xmlns:wsoap='http://schemas.xmlsoap.org/wsdl/soap/'
(25)                  xmlns:xs='http://www.w3.org/2001/XMLSchema'>
(26)          <wsdl:import
(27)              namespace='http://services.example.org/stockquote'
(28)              location='http://services.example.org/stockquote/schemas' />
(29)          <wsdl:portType name='StockQuotePortType'>
(30)              <wsdl:operation name='GetLastTradePrice'>
(31)                  <wsdl:input message='msgs:GetLastTradePriceInput'
(32)                      name='GetLastTradePriceInput' />
(33)                  <wsdl:output message='msgs:GetLastTradePriceOutput'
(34)                      name='GetLastTradePriceOutput' />
(35)              </wsdl:operation>
(36)          </wsdl:portType>
(37)          <wsdl:binding name='StockQuoteBinding'
(38)              type='tns:StockQuotePortType'>
(39)              <wsp:PolicyReference
(40)                  URI='http://services.example.org/stockquote/policy' />
(41)              <wsoap:binding style='document'
(42)                  transport='http://schemas.xmlsoap.org/soap/http' />
(43)              <wsdl:operation name='GetLastTradePrice'>
(44)                  <wsoap:operation
(45)                      soapAction='http://services.example.org/stockquote/GetLastTradePrice' />
(46)                      <wsdl:input name='GetLastTradePriceInput'>
(47)                          <wsoap:body use='literal' />
(48)                      </wsdl:input>
(49)                      <wsdl:output name='GetLastTradePriceOutput'>
(50)                          <wsoap:body use='literal' />
(51)                      </wsdl:output>
(52)                  </wsdl:operation>
(53)              </wsdl:binding>
(54)              <wsdl:service name='StockQuoteService'>
(55)                  <wsdl:port name='StockQuotePort'>
(56)                      binding='tns:StockQuoteBinding' >
(57)                      <wsoap:address
(58)                          location='http://services.example.org/stockquote' />
(59)                  </wsdl:port>
(60)              </wsdl:service>
(61)          </mex:MetadataSection>
(62)          <mex:MetadataSection
(63)              Dialect='http://www.w3.org/2001/XMLSchema'
(64)              Identifier='http://services.example.org/stockquote/schemas'>
(65)              <mex:Location>

```

```

(66)          http://services.example.org/stockquote/schemas
(67)          </mex:Location>
(68)          </mex:MetadataSection>
(69)          <mex:MetadataSection
(70)              Dialect='http://schemas.xmlsoap.org/ws/2004/09/policy'
(71)              Identifier='http://services.example.org/stockquote/policy'>
(72)              <mex:MetadataReference>
(73)                  <wsa10:Address>
(74)                      http://services.example.org/stockquote/policy
(75)                  </wsa10:Address>
(76)              </mex:MetadataReference>
(77)          </mex:MetadataSection>
(78)      </mex:Metadata>
(79)  </s11:Body>
(80) </s11:Envelope>

```

The message of Table 2 is a [[WS-Transfer](#)] response message to the request of Table 1. The content of the [Body] (lines 16-79) is a mex:Metadata element with metadata about the Web service endpoint (lines 17-78). The mex:Metadata contains three Metadata Sections. The first Metadata Section (lines 18-61) contains the WSDL [[WSDL](#)] of the Web service endpoint. The second Metadata Section (lines 62-68) contains the location of the XML Schemas [[XML Schema Part 1](#), [Part 2](#)] used by the WSDL document. The schemas can be retrieved through an HTTP GET request at the identified URL (lines 65-67). The third Metadata Section (lines 69-77) contains the WS-Addressing [[WS-Addressing 1.0 Core](#)] endpoint reference (lines 72-75) of a WS-Transfer [[WS-Transfer](#)] resource the representation of which is a WS-Policy [[WS-Policy](#)] document as indicated by the Dialect attribute (line 70). The WS-Policy document is the same as the one indicated in the WSDL document (lines 39-40).

While the WS-Policy of the Web service endpoint could be retrieved using a WS-Transfer GET request directed to the endpoint identified by the mex:MetadataReference element in lines 72-76 of Table 2, some endpoints may choose to support explicit request for metadata. Table 3 illustrates a sample Get Metadata request for the WS-Policy [[WS-Policy](#)].

**Table 3: Sample Get Metadata request message**

```

(01) <s11:Envelope
(02)     xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
(03)     xmlns:wsa10='http://www.w3.org/2005/08/addressing'
(04)     xmlns:mex='http://schemas.xmlsoap.org/ws/2004/09/mex' >
(05)     <s11:Header>
(06)         <wsa10:To>http://services.example.org/stockquote</wsa10:To>
(07)         <wsa10:Action>
(08)             http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request
(09)         </wsa10:Action>
(10)         <wsa10:MessageID>
(11)             urn:uuid:73d7edfc-5c3c-49b9-ba46-2480caee43e9
(12)         </wsa10:MessageID>
(13)         <wsa10:ReplyTo>
(14)             <wsa10:Address>http://client.example.org</wsa10:Address>
(15)         </wsa10:ReplyTo>
(16)     </s11:Header>
(17)     <s11:Body>

```

```

(18)    <mex:GetMetadata>
(19)
    <mex:Dialect>http://schemas.xmlsoap.org/ws/2004/09/policy</mex:Dialect>
(20)    <mex:Identifier>
(21)        http://services.example.org/stockquote/policy
(22)    </mex:Identifier>
(23)    </mex:GetMetadata>
(24)  </s11:Body>
(25) </s11:Envelope>

```

Lines 7-9 in Table 3 indicate this is a Get Metadata request. As lines 18-22 indicate, this request is for the policy of the Web service endpoint (line 6).

Table 4 lists a sample response to the request in Table 3.

**Table 4: Sample Get Metadata response message**

```

(01) <s11:Envelope
(02)     xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
(03)     xmlns:wsa10='http://www.w3.org/2005/08/addressing'
(04)     xmlns:wsp='http://schemas.xmlsoap.org/ws/2004/09/policy'
(05)     xmlns:mex='http://schemas.xmlsoap.org/ws/2004/09/mex'>
(06) <s11:Header>
(07)     <wsa10:To>http://client.example.org</wsa10:To>
(08)     <wsa10:Action>
(09)         http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
(10)     </wsa10:Action>
(11)     <wsa10:RelatesTo>
(12)         urn:uuid:73d7edfc-5c3c-49b9-ba46-2480caee43e9
(13)     </wsa10:RelatesTo>
(14) </s11:Header>
(15) <s11:Body>
(16)     <mex:Metadata>
(17)         <mex:MetadataSection
(18)             Dialect='http://schemas.xmlsoap.org/ws/2004/09/policy'
(19)             Identifier='http://services.example.org/stockquote/policy'>
(20)             <wsp:Policy>
(21)                 <wsp:ExactlyOne>
(22)                     <!-- Policy alternatives -->
(23)                 </wsp:ExactlyOne>
(24)             </wsp:Policy>
(25)         </mex:MetadataSection>
(26)     </mex:Metadata>
(27) </s11:Body>
(28) </s11:Envelope>

```

Lines 8-10 in Table 4 indicate this message is a response to a Get Metadata request, and lines 11-13 indicate that it is a response to the request in Table 3. Lines 16-26 contain a single Metadata Section (lines 17-25); line 18 indicates that the metadata in this section is of type, or dialect, WS-Policy while line 19 identifies a specific policy document. Line 22 would have contained the policy expressions for the Web service endpoint to which the Get Metadata request of Table 3 was directed.

## 2. Notation

### 2.1 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 5 lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

**Table 5: 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	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>	SOAP 1.1 [ <a href="#">SOAP 1.1</a> ]
s12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>	SOAP 1.2 [ <a href="#">SOAP 1.2</a> ]
wsa04	<a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a>	WS-Addressing 2004 [ <a href="#">WS-Addressing 2004</a> ]
wsa10	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>	WS-Addressing 1.0 Core [ <a href="#">WS-Addressing 1.0 Core</a> ]
wsdl	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	WSDL [ <a href="#">WSDL 1.1</a> ]
wsp	<a href="http://schemas.xmlsoap.org/ws/2004/09/policy">http://schemas.xmlsoap.org/ws/2004/09/policy</a>	WS-Policy [ <a href="#">WS-Policy</a> ]
mex	<a href="http://schemas.xmlsoap.org/ws/2004/09/mex">http://schemas.xmlsoap.org/ws/2004/09/mex</a>	This specification
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	XML Schema [ <a href="#">Part 1, 2</a> ]
wxf	<a href="http://schemas.xmlsoap.org/ws/2004/09/transfer">http://schemas.xmlsoap.org/ws/2004/09/transfer</a>	WS-Transfer [ <a href="#">WS-Transfer</a> ]

### 2.2 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 outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of literal 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.

- The characters "[" and "]" are used to call out references and property names.
- Ellipses (i.e., "...") indicate points of extensibility. 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. By default, if a receiver does not recognize an extension, the receiver SHOULD ignore the extension; exceptions to this processing rule, if any, are clearly indicated below.
- XML namespace prefixes (see Table 5) are used to indicate the namespace of the element being defined.

In addition to Message Information Header properties [[WS-Addressing 2004](#)] and Message Addressing Properties [[WS-Addressing 1.0 Core](#)], this specification uses the following properties to define messages:

#### **[Headers]**

Unordered message headers.

#### **[Body]**

A message body.

These properties bind to a SOAP 1.1 Envelope [[SOAP 1.1](#)] as follows:

```
<s11:Envelope>
  <s11:Header>[Headers] ... </s11:Header>
  <s11:Body>[Body]</s11:Body>
</s11:Envelope>
```

These properties bind to a SOAP 1.2 Envelope [[SOAP 1.2](#)] as follows:

```
<s12:Envelope>
  <s12:Header>[Headers] ... </s12:Header>
  <s12:Body>[Body]</s12:Body>
</s12:Envelope>
```

### **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 XML documents, SOAP Envelopes, and [[WS-Addressing 2004](#), [WS-Addressing 1.0 Core](#)] endpoint references unless it is compliant with this specification.

Support for the GetMetadata operation by a Web service is optional. If metadata about a Web service endpoint is referenced by a Metadata Reference, which is a [[WS-Addressing 2004](#), [WS-Addressing 1.0 Core](#)] endpoint reference, then the Metadata Reference MUST refer to a [[WS-Transfer](#)] resource. The referred resource MAY also support other resource management and access specifications (e.g. HTTP, WS-ResourceFramework).

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

### **3. Metadata Resources**

A resource is a Web service that is addressable by an endpoint reference [[WS-Addressing 2004](#), [WS-Addressing 1.0 Core](#)] and can be represented by an XML

InfoSet. The resource's representation can be retrieved using the Get operation defined in [\[WS-Transfer\]](#).

When the representation of a resource is mex:Metadata, as defined in Section 4, or any other document format (e.g. [\[XML Schema Part 1, Part 2\]](#), [\[WSDL\]](#), [\[WS-Policy\]](#)) for which a mex:MetadataSection/@Dialect has been defined, then the resource is referred as 'metadata resource'. The representation of a metadata resource MAY be retrieved and/or updated as any other [\[WS-Transfer\]](#) resource. Specifically, the representation of a metadata resource MUST be retrievable through a [\[WS-Transfer\]](#) Get operation.

A Web service endpoint MAY have one or more associated metadata resources. A Web service endpoint MAY also support direct retrieval of metadata by requesters using a GetMetadata operation directed to the Web service endpoint itself, as described in Section 5.2.

A metadata resource MAY support other operations defined by [\[WS-Transfer\]](#), such as Put (e.g. to allow update of non-static metadata by authorized agents), or other resource management and access specifications (e.g. HTTP, WS-ResourceFramework).

## 4. Web Services Metadata

The Web service Metadata element is a collection of metadata units such as [\[WSDL\]](#) definitions, XML Schema [\[XML Schema Part 1, Part 2\]](#) documents, [\[WS-Policy\]](#) expressions, etc. Each unit corresponds to metadata for a given scope, domain, or namespace. The collection of units is represented by a Metadata element, and units within the collection are represented by Metadata Section elements.

To facilitate processing, Metadata Sections are tagged with a @Dialect and (optionally) @Identifier of the metadata unit. To ensure scalability, a unit of metadata may be included in-line within its Metadata Section or may be included by reference, either an endpoint reference [\[WS-Addressing 2004\]](#), [\[WS-Addressing 1.0 Core\]](#) (Metadata Reference) or a URL (Location).

The outline for the Metadata element is:

```
<mex:Metadata ...>
  <mex:MetadataSection Dialect="xs:anyURI"
    (Identifier="xs:anyURI")? ...>
  (
    <mex:MetadataReference ...>
      endpoint-reference
    </mex:MetadataReference>
    |
    <mex:Location>xs:anyURI</mex:Location>
    |
    DialectSpecificElement
  )
  </mex:MetadataSection>*
  ...
</mex:Metadata>
```

The following describes additional constraints on the outline listed above:

/mex:Metadata

This contains one Metadata Section child for each distinct unit of metadata. When there is a large amount of metadata, the [children] SHOULD contain Metadata References or Locations instead of the actual information.

/mex:Metadata/mex:MetadataSection

This represents a single unit of metadata (e.g., a WSDL definitions, an XML Schema document) for a given scope, domain, or namespace. A Metadata Section contains exactly one child element, either the embedded XML for the metadata unit, an endpoint reference to a Metadata Resource for the metadata unit, or a URL to metadata unit.

/mex:Metadata/mex:MetadataSection/@Dialect

This indicates the format and version of the metadata unit contained in this Metadata Section (e.g., WSDL version 1.1). Dialect is an absolute URI. This value should be compared directly, as a case-sensitive string, with no attempt to unescape or to otherwise canonicalize it.

This specification defines the following values for Dialect; other specifications should define values for Dialect for their metadata format(s).

Dialect URI	Metadata Format
http://www.w3.org/2001/XMLSchema	xs:schema [ <a href="#">XML Schema Part 1</a> ]
http://schemas.xmlsoap.org/wsdl/	wsdl:definitions [ <a href="#">WSDL 1.1</a> ]
http://schemas.xmlsoap.org/ws/2004/09/policy	wsp:Policy [ <a href="#">WS-Policy</a> ]
http://schemas.xmlsoap.org/ws/2004/09/policy/attachment	wsp:PolicyAttachment [ <a href="#">WS-PolicyAttachment</a> ]
http://schemas.xmlsoap.org/ws/2004/09/mex	mex:Metadata [This specification]

If there is more than one metadata unit with the same Dialect, e.g., more than one XML Schema document, including them all, one per Metadata Section, is explicitly encouraged.

/mex:Metadata/mex:MetadataSection/@Dialect

= "http://schemas.xmlsoap.org/ws/2004/09/mex"

This value indicates the type of the metadata contained within the Metadata Section. When used in conjunction with Metadata Reference or Location, this allows the inclusion of metadata by reference.

/mex:Metadata/mex:MetadataSection/@Identifier

This indicates the Identifier for the metadata unit in this Metadata Section. Identifier is an absolute URI. This value should be compared directly, as a case-sensitive string, with no attempt to unescape or to otherwise canonicalize it. If omitted, there is no implied value.

The interpretation of Identifier is Dialect-specific. While the Dialect attribute indicates the metadata format and version of the metadata in the Metadata Section, the Identifier attribute MAY be used to identify a Metadata Section or to just differentiate between Metadata Sections containing the same type of metadata. The value of the Identifier MAY be the same as the value of one of the attributes or elements of the metadata in the Metadata Section (if the metadata is included by value) or in the representation of a metadata resource (if the metadata is include by reference through Metadata Reference or Location). A metadata publisher MAY choose any value for the Identifier. The values of Identifier attributes in multiple Metadata Section elements in the same Metadata element MAY be the same. For well-known metadata formats, it is RECOMMENDED that the value of the Identifier comes from the metadata when that is possible, as the table below shows.

Dialect URI	@Identifier value
http://www.w3.org/2001/XMLSchema	xs:schema/@targetNamespace
http://schemas.xmlsoap.org/wsdl/	wsdl:definitions/@targetNamespace
http://schemas.xmlsoap.org/ws/2004/09/policy	wsp:Policy/@Name
http://schemas.xmlsoap.org/ws/2004/09/policy/attachment	Not defined
http://schemas.xmlsoap.org/ws/2004/09/mex	Not defined

If there is more than one metadata section with the same identifier, e.g., more than one XML Schema in the same target namespace, including them all, one per metadata section, is explicitly encouraged.

#### /mex:Metadata/mex:MetadataSection/mex:MetadataReference

This is an endpoint reference to a metadata resource and is of type EndpointReferenceType as defined by [[WS-Addressing 2004](#)] or EndpointReferenceType as defined by [[WS-Addressing 1.0 Core](#)]. The resource MUST support the GET operation [[WS-Transfer](#)] to allow the retrieval of the metadata unit for the Metadata Section's Dialect and Identifier (if any). When this element is present, it MUST have no element siblings.

#### /mex:Metadata/mex:MetadataSection/mex:Location

This contains a URL to metadata, and the metadata MUST be retrievable from that URL using the primary access mechanism for the scheme of the URL. For example, 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.

#### /mex:Metadata/mex:MetadataSection/*DialectSpecificElement*

When any element other than Metadata Reference or Location is present, the element is to be interpreted as the representation of the metadata unit associated with the Metadata Section's Dialect and Identifier.

## 5. Retrieving Metadata

### 5.1 WS-Transfer Get

To retrieve the representation of a metadata resource, a requester MAY send a WS-Transfer [[WS-Transfer](#)] Get request message to the metadata resource's endpoint. The WS-Transfer Get request fetches a one-time snapshot of the metadata. The metadata associated with a service endpoint may be available as multiple metadata resources. As a result, the metadata returned by the Get request to a metadata resource's endpoint may be limited to a particular metadata type (@Dialect) and identifier (@Identifier).

The representation of a metadata resource MAY be a mex:Metadata element or any other document format (e.g. [[XML Schema Part 1, Part 2](#)], [[WSDL](#)], [[WS-Policy](#)]) for which a mex:MetadataSection/@Dialect has been defined.

### 5.2 Get Metadata

When the metadata for an endpoint is not available or is unknown and there is no information on how to retrieve it (e.g. an endpoint reference to a [[WS-Transfer](#)] resource representing the metadata), a requester MAY send a Get Metadata request message to that endpoint to retrieve its metadata. A service endpoint MAY support the Get Metadata request. Observe that, in the case where a service endpoint is also a [[WS-Transfer](#)] resource, the [[WS-Transfer](#)] Get operation allows requesters to retrieve the resource representation associated with that endpoint (i.e. the "data").

while the GetMetadata operation can be used to retrieve that endpoint's metadata. The normative outline for a Get Metadata request is:

```
[action]
http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request

[Body]
<mex:GetMetadata ...>
  (<mex:Dialect>xs:anyURI</mex:Dialect>
   (<mex:Identifier>xs:anyURI</mex:Identifier>)??
  )?
</mex:GetMetadata>
```

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

[Body]/mex:GetMetadata/mex: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, there is no implied value and so the response may include Metadata Sections with any dialect.

[Body]/mex:GetMetadata/mex: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 ./mex:Dialect is present. If multiple Metadata Sections have the indicated Dialect and Identifier then all of them MUST be returned.

Other message information headers defined by WS-Addressing [[WS-Addressing 2004](#), [WS-Addressing 1.0 Core](#)] 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 2004](#), [WS-Addressing 1.0 Core](#)] (e.g., wsa04: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:

```
[action]
http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response

[Body]
<mex:Metadata ...>
  ...
</mex:Metadata>
```

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

[Body]/mex:Metadata

The body of the response message MUST contain one mex:Metadata element as defined in Section 4 of this specification.

## 6. Metadata in Endpoint References

Section 5 describes two mechanisms that allow requesters to retrieve (“pull metadata”) Web services metadata from an endpoint (GetMetadata) or a metadata resource (WS-Transfer Get).

In addition to these mechanisms, the issuer of a service endpoint reference MAY include Web services metadata for that service inside the endpoint reference itself. This mechanism (“push metadata”) simplifies the bootstrapping of the service interaction on the requester side by avoiding additional calls to retrieve (pull) the Web service metadata for the service endpoint.

Embedding a mex:Metadata element inside an endpoint reference allows the issuer of the endpoint reference to include metadata by value or by reference, according to the options described in Section 4. The outline of a [[WS-Addressing 2004](#)] endpoint reference which includes Web services metadata is as follows:

```
<wsa04:EndpointReference>
    <wsa04:Address>xs:anyURI</wsa04:Address>
    <wsa04:ReferenceProperties>...</wsa04:ReferenceProperties> ?
    <wsa04:ReferenceParameters>...</wsa04:ReferenceParameters> ?
    <wsa04:PortType>xs:QName</wsa04:PortType> ?
    <wsa04:ServiceName PortName="xs:NCName"?>xs:QName</wsa04:ServiceName> ?
    <wsp:Policy>...</wsp:Policy>*
    <mex:Metadata>...</mex:Metadata>?
    ...
</wsa04:EndpointReference>
```

/wsa04:EndpointReference/mex:Metadata

A unique mex:Metadata element appearing as a direct child of the endpoint reference element.

The outline of a [[WS-Addressing 1.0 Core](#)] endpoint reference which includes Web services metadata is as follows:

```
<wsa10:EndpointReference>
    <wsa10:Address>xs:anyURI</wsa10:Address>
    <wsa10:ReferenceParameters>...</wsa10:ReferenceParameters> ?
    <wsa10:Metadata>
        <mex:Metadata>...</mex:Metadata>?
        ...
    </wsa10:Metadata> ?
    ...
</wsa10:EndpointReference>
```

/wsa10:EndpointReference/wsa10:Metadata/mex:Metadata

A unique mex:Metadata element appearing as a direct child of the wsa10:Metadata endpoint reference element.

In the example of Table 6, a [[WS-Addressing 1.0 Core](#)] endpoint reference contains WSDL metadata by value that identifies the binding used to access the service endpoint:

**Table 6. Endpoint reference with embedded WSDL metadata**

```
(01) <wsa10:EndpointReference
(02)     xmlns:wsa10='http://www.w3.org/2005/08/addressing'>
(03)     <wsa10:Address>http://services.example.org/stockquote</wsa10:Address>
(04)     <wsa10:Metadata>
```

```

(05)    <mex:Metadata xmlns:mex='http://schemas.xmlsoap.org/ws/2004/09/mex'>
(06)        <mex:MetadataSection Dialect='http://schemas.xmlsoap.org/wsdl/'>
(07)            <wsdl:definitions
(08)                name='StockQuote'
(09)                targetNamespace='http://services.example.org/stockquote'
(10)                xmlns:tns='http://services.example.org/stockquote'
(11)                xmlns:wsoap='http://schemas.xmlsoap.org/wsdl/soap/'
(12)                xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
(13)                xmlns:wpolicy='http://schemas.xmlsoap.org/ws/2004/09/policy'
(14)                xmlns:msgs='http://services.example.org/stockquote/schemas'
(15)                xmlns:xs='http://www.w3.org/2001/XMLSchema'>
(16)        <wsdl:import
(17)            namespace='http://services.example.org/stockquote'
(18)            location='http://services.example.org/stockquote/schemas' />
(19)        <wsdl:portType name='StockQuotePortType'>
(20)            <wsdl:operation name='GetLastTradePrice'>
(21)                <wsdl:input message='msgs:GetLastTradePriceInput'
(22)                    name='GetLastTradePriceInput' />
(23)                <wsdl:output message='msgs:GetLastTradePriceOutput'
(24)                    name='GetLastTradePriceOutput' />
(25)            </wsdl:operation>
(26)        </wsdl:portType>
(27)        <wsdl:binding name='StockQuoteBinding'
(28)            type='tns:StockQuotePortType'>
(29)            <wsp:PolicyReference
(30)                URI='http://services.example.org/stockquote/policy' />
(31)            <wsoap:binding style='document'
(32)                transport='http://schemas.xmlsoap.org/soap/http' />
(33)            <wsdl:operation name='GetLastTradePrice'>
(34)                <wsoap:operation
(35)                    soapAction='http://services.example.org/stockquote/GetLastTradePrice' />
(36)                    <wsdl:input name='GetLastTradePriceInput'>
(37)                        <wsoap:body use='literal' />
(38)                    </wsdl:input>
(39)                    <wsdl:output name='GetLastTradePriceOutput'>
(40)                        <wsoap:body use='literal' />
(41)                    </wsdl:output>
(42)                </wsdl:operation>
(43)            </wsdl:binding>
(44)            <wsdl:service name='StockQuoteService'>
(45)                <wsdl:port name='StockQuotePort'
(46)                    binding='tns:StockQuoteBinding' >
(47)                    <wsoap:address
(48)                        location='http://services.example.org/stockquote' />
(49)                    </wsdl:port>
(50)                </wsdl:service>
(51)            </wsdl:definitions>
(52)        </mex:MetadataSection>

```

```

(53)      Dialect='http://www.w3.org/2001/XMLSchema'
(54)      Identifier='http://services.example.org/stockquote/schemas'
(55)      <mex:MetadataReference>
(56)          <wsa10:Address>
(57)              http://services.example.org/stockquote/schemas
(58)          </wsa10:Address>
(59)      </mex:MetadataReference>
(60)      </mex:MetadataSection>
(61)  </mex:Metadata>
(62) </wsa10:Metadata>
(63) </wsa10:EndpointReference>

```

Lines 4-62 in Table 6 show the way a Metadata element can be embedded in the endpoint reference of a service. The Metadata element contains two Metadata Sections. In the first one (lines 6-51) the WSDL of the Web service endpoint is included 'by value'. In the second Metadata Section (lines 52-60) an endpoint reference to a metadata resource is shown. The representation of the metadata resource is an XML Schema as hinted by the Dialect attribute (line 53) and is distinguished from other XML Schemas for that Web service endpoint by the value of Identifier attribute (line 54) which in this case happens to be the same as the value of the targetNamespace attribute of the XML Schema. The [WS-Transfer] Get operation can be used to retrieve the referred XML Schema. Since no metadata is available about the metadata resource, it is assumed that binding-related information was communicated out-of-band (e.g. an application domain-specific specification has defined a binding for that domain) so that the requester could send a [WS-Transfer] Get request to the metadata resource's endpoint, as defined in Section 7.

**Table 7. Endpoint reference with embedded metadata about the Metadata Reference**

```

(01) <wsa10:EndpointReference
(02)     xmlns:wsa10='http://www.w3.org/2005/08/addressing'>
(03)     <wsa10:Address>http://services.example.org/stockquote</wsa10:Address>
(04)     <wsa10:Metadata>
(05)         <mex:Metadata xmlns:mex='http://schemas.xmlsoap.org/ws/2004/09/mex'>
(06)             <mex:MetadataSection
(07)                 Dialect='http://schemas.xmlsoap.org/ws/2004/09/mex'>
(08)                 <mex:MetadataReference>
(09)                     <wsa10:Address>
(10)                         http://services.example.org/stockquote/metadata
(11)                     </wsa10:Address>
(12)                 <wsa10:Metadata>
(13)                     <mex:Metadata>
(14)                         <mex:MetadataSection
(15)                             Dialect='http://schemas.xmlsoap.org/wsdl/'>
(16)                             <!--Reference to WS-Transfer WSDL wxf:Resource portType
(17)                                 plus binding -->
(18)                         </mex:MetadataSection>
(19)                     </mex:Metadata>
(20)                 </wsa10:Metadata>
(21)             </mex:MetadataReference>
(22)         </mex:MetadataSection>

```

```
(23)    </mex:Metadata>
(24)  </wsa10:Metadata>
(25) </wsa10:EndpointReference>
```

Table 7 shows an example of a Web service endpoint reference in which the Metadata element contains a single Metadata Reference element to a metadata resource (lines 8-21) the representation of which is mex:Metadata as hinted by the value of the Dialect attribute (line 7). The Metadata Reference contains a Metadata element (lines 13-19) which contains the necessary metadata for interacting with the metadata resource through [WS-Transfer] operations. Since the Metadata Reference is an endpoint reference to a [WS-Transfer] resource, the embedded metadata includes the [WS-Transfer] WSDL portType and the necessary binding information for communicating with that resource.

## 7. Bootstrapping Metadata Retrieval

This specification provides several mechanisms to aid service endpoints and service requesters in bootstrapping the interaction. In particular, the mechanisms described in Section 6 allow issuers of endpoint references to include sufficient protocol binding information to allow requesters to issue a Get request against a metadata resource, or a GetMetadata request against a service endpoint, in order to retrieve all the information needed to use the service.

When that information is not available, however, requesters must rely on contextual or out-of-band information in order to choose the protocol binding most appropriate to use in retrieving Web service metadata. Furthermore, specific protocol bindings for metadata retrieval may be defined by communities within the context of particular application domains.

## 8. Security

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 2004, WS-Addressing 1.0 Core], 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 [[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.

## 9. Acknowledgements

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

## 10. References

### [RFC 2119]

S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997. (See <http://www.ietf.org/rfc/rfc2119.txt>.)

### [SOAP 1.1]

D. Box, et al, "Simple Object Access Protocol (SOAP) 1.1," May 2000. (See [http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.](http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.))

#### [SOAP 1.2]

M. Gudgin, et al, "SOAP Version 1.2 Part 1: Messaging Framework," June 2003. (See [http://www.w3.org/TR/2003/REC-soap12-part1-20030624/.](http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>.))

#### [WS-Addressing 2004]

D. Box, et al, "Web Services Addressing (WS-Addressing)," August 2004. (See [http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/.](http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>.))

#### [WS-Addressing 1.0 Core]

M. Gudgin, M. Hadley, T. Rogers (editors), "Web Services Addressing 1.0 – Core" (Proposed Recommendation), March 2006. (See [http://www.w3.org/TR/2006/PR-ws-addr-core-20060321/.](http://www.w3.org/TR/2006/PR-ws-addr-core-20060321/>.))

#### [WS-Policy]

S. Bajaj, et al, "Web Services Policy Framework (WS-Policy)," March 2006. (See [http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/.](http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/>.))

#### [WS-PolicyAttachment]

S. Bajaj, et al, "Web Services Policy Attachment (WS-PolicyAttachment)," March 2006. (See [http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/.](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/>.))

#### [WS-SecureConversation]

S. Anderson, et al, "Web Services Secure Conversation Language (WS-SecureConversation)," February 2005. (See <http://www.oasis-open.org/committees/download.php/17364.>)

#### [WS-Security]

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://www.oasis-open.org/committees/download.php/16569/.>)

#### [WS-Transfer]

J. Alexander, et al, "Web Services Transfer (WS-Transfer)," September 2004. (See [http://www.w3.org/Submission/2006/SUBM-WS-Transfer-20060315/.](http://www.w3.org/Submission/2006/SUBM-WS-Transfer-20060315/>.))

#### [WS-Trust]

S. Anderson, et al, "Web Services Trust Language (WS-Trust)," February 2005. (See <http://www.oasis-open.org/committees/download.php/17364.>)

#### [WSDL 1.1]

E. Christensen, et al, "Web Services Description Language (WSDL) 1.1," March 2001. (See [http://www.w3.org/TR/2001/NOTE-wsdl-20010315/.](http://www.w3.org/TR/2001/NOTE-wsdl-20010315.))

#### [XML Schema, Part 1]

H. Thompson, et al, "XML Schema Part 1: Structures," October 2004. (See [http://www.w3.org/TR/2004/REC-xmleschema-1-20041028/.](http://www.w3.org/TR/2004/REC-xmleschema-1-20041028/>.))

#### [XML Schema, Part 2]

P. Biron, et al, "XML Schema Part 2: Datatypes," October 2004. (See [http://www.w3.org/TR/2004/REC-xmleschema-2-20041028/.](http://www.w3.org/TR/2004/REC-xmleschema-2-20041028/>.))

## Appendix I – XML Schema

A normative copy of the XML Schema [[XML Schema Part 1](#), [Part 2](#)] description for this specification may be retrieved from the following address:

<http://schemas.xmlsoap.org/ws/2004/09/mex/MetadataExchange.xsd>

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:wsa10='http://www.w3.org/2005/08/addressing'
  xmlns:wsa04='http://schemas.xmlsoap.org/ws/2004/08/addressing'
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  elementFormDefault='qualified'
  blockDefault='#all' >

  <!-- 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:any namespace='##other' processContents='lax'
          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' />
        <xs:element ref='tns:MetadataReference' />
      </xs:choice>
    </xs:complexType>
  </xs:element>
```

```

        <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>

<!--
    Ideally, the type of the MetadataReference would have been
    the union of wsa04:EndpointReferenceType and
    wsa10:EndpointReferenceType but unfortunately xs:union only
    works for simple types. As a result, we have to define
    the mex:MetadataReference using xs:any.
-->

<xs:element name='MetadataReference'>
    <xs:complexType>
        <xs:sequence>
            <xs:any minOccurs='1' maxOccurs='unbounded'
                    processContents='lax' namespace='##other' />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name='Location'
            type='xs:anyURI' />
</xs:schema>
```

## Appendix II – 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:wsa10='http://www.w3.org/2005/08/addressing'
    xmlns:wsa04='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:portType name='MetadataExchange' >
  <wsdl:operation name='GetMetadata' >
    <wsdl:input
      message='tns:GetMetadataMsg'
      wsdl:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request'
      wsdl:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Request' />
    <wsdl:output
      message='tns:GetMetadataResponseMsg'
      wsdl:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response'
      wsdl:Action=
        'http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response' />
  </wsdl:operation>
</wsdl:portType>

</wsdl:definitions>
```