WSDL 1.1 Binding Extension for SOAP 1.2

March 2, 2006

Authors

Dimitar Angelov, SAP
Keith Ballinger, Microsoft
Russell Butek, IBM
Doug Davis, IBM
Christopher Ferris (Editor), IBM
Anish Karmarkar, Oracle
Canyang Kevin Liu, SAP
Jonathan Marsh (Editor), Microsoft
Jeff Mischkinsky, Oracle
Jeffrey Schlimmer, Microsoft
Ümit Yalçınalp, SAP

Copyright Notice

Abstract

This specification defines WSDL 1.1 binding extensions to indicate that Web service messages are bound to the SOAP 1.2 protocol.

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 warrantees or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction

This specification defines WSDL 1.1 binding extensions to indicate that Web service messages are bound to the SOAP 1.2 protocol.

1.1 Requirements

This specification intends to meet the following requirements:

- Provide functionality comparable to binding for SOAP 1.1 [WSDL 1.1] for SOAP 1.2 [SOAP 1.2 Part 1, Part 2]. Specifically:
    - Indicate that a binding is bound to the SOAP 1.2 protocol.
    - Specify an address for a SOAP endpoint.
    - Specify the URI for the action parameter of the application/soap+xml Content-Type HTTP header value [SOAP Media] for the HTTP binding of SOAP.
    - Define Headers that are transmitted as part of the SOAP Envelope.
- Indicate whether an action parameter is required by a SOAP 1.2 endpoint.
- Provide extensibility for more sophisticated and/or currently unanticipated scenarios.

1.2 Example

Table 1 lists an example Web service description bound to SOAP 1.2.

Table 1: Example SOAP 1.2 binding.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
  targetNamespace="http://example.com"
  xmlns:tns="http://example.com"
  xmlns:wsoap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" >

  <wsdl:types>
    <xs:schema
      targetNamespace="http://example.com"
      blockDefault="#all"
      elementFormDefault="qualified" >
      <xs:element name="HelloResponse" type="xs:string" />
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="HelloWorldMessageIn" />

  <wsdl:message name="HelloWorldMessageOut">
    <wsdl:part name="parameters" element="tns:HelloResponse"/>
  </wsdl:message>

  <wsdl:portType name="Test">
    <wsdl:operation name="HelloWorld">
```

```
(26)       <wsdl:input message="tns:HelloWorldMessageIn" />
(27)       <wsdl:output message="tns:HelloWorldMessageOut" />
(28)     </wsdl:operation>
(29)   </wsdl:portType>
(30)
(31)   <wsdl:binding name="TestSoap12Binding" type="tns:Test">
(32)     <wsoap12:binding
(33)       transport="http://schemas.xmlsoap.org/soap/http "
(34)       style="document" />
(35)     <wsdl:operation name="HelloWorld">
(36)       <wsoap12:operation
(37)         soapAction="http://example.com/Test/HelloWorldRequest"
(38)         soapActionRequired="true" />
(39)       <wsdl:input>
(40)         <wsoap12:body use="literal" />
(41)       </wsdl:input>
(42)       <wsdl:output>
(43)         <wsoap12:body use="literal" />
(44)       </wsdl:output>
(45)     </wsdl:operation>
(46)   </wsdl:binding>
(47)
(48)   <wsdl:service name="HelloWorld">
(49)     <wsdl:port name="HelloWorldSoap12"
(50)                binding="tns:TestSoap12Binding">
(51)       <wsoap12:address
(52)         location="http://localhost/helloworld" />
(53)     </wsdl:port>
(54)   </wsdl:service>
(55)
(56) </wsdl:definitions>
(57)
```

Lines (31-46) in Table 1 are a binding of the port type in Lines (24-29). The extension in Lines (32-34) is in the namespace of this specification (Line 05) and indicates the messages use the SOAP 1.2 [SOAP 1.2] protocol, over HTTP (Line 33), and using the document style (Line 34).

Lines (36-38) indicates the value of the action parameter (Line 37) and that the service requires this parameter (Line 38).

Line (40) and Line (43) indicate the SOAP Body is literally described. Specifically, Line (43) indicates that the SOAP Body of the response is described the XML Schema element declaration in Line (14).

Lines (49-53) are a port bound to the binding in Lines (31-46). The binding extension in Lines (51-52) indicates the service may be reached at "http://localhost/helloworld".

2. Terminology and Notation

2.1 XML Namespaces

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

```
http://schemas.xmlsoap.org/wsdl/soap12/
```

Table 2 lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 2: Prefixes and XML Namespaces used in this specification.

Prefix	XML Namespace	Specification(s)
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL 1.1]
wsoap12	http://schemas.xmlsoap.org/wsdl/soap12/	This specification
xs	http://www.w3.org/2001/XMLSchema	[XML Schema Part 1, Part 2]
soap	http://www.w3.org/2003/05/soap-envelope	[SOAP 1.2]

The normative schema for the SOAP 1.2 Binding Extension for WSDL 1.1 can be found at:

```
http://schemas.xmlsoap.org/wsdl/soap12/wsdl11soap12.xsd
```

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].

Namespace URIs of the general form "http://www.example.com/…" represents some application-dependent or context-dependent URI as defined in RFC2396 [URI].

The characters "[" and "]" are used to call out references and XML Information Set [Infoset] property names.

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.
- 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 2) are used to indicate the namespace of the element being defined.
- Grammar in bold has not been introduced earlier in the document, or is of particular interest in an example.
- Examples starting with `<?xml` contain enough information to conform to this specification; other examples are fragments and require additional information to be specified in order to conform.

2.3 Compliance

An endpoint MAY implement more than one of the roles defined herein. An endpoint is not compliant with this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined herein for the roles it implements.

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. SOAP 1.2 Binding

There are three key differences from the SOAP 1.1 binding extensions [WSDL 1.1]:

- A new namespace. (See Section 2.1.)
- The encodingStyle attribute is now a single URI, instead of a list of URIs.
- There is a new attribute: soapActionRequired, which is used to indicate that the server needs the action parameter value.

The outline for the SOAP 1.2 binding extensions are:

```
<wsdl:definitions ...>

  ...

  <wsdl:binding ...>
    <wsoap12:binding style="rpc|document" ?
                     transport="xs:anyURI"
                     wsdl:required="xs:boolean" ? />
    <wsdl:operation ...>
      <wsoap12:operation soapAction="xs:anyURI" ?
                         soapActionRequired="xs:boolean" ?
                         style="rpc|document" ?
                         wsdl:required="xs:boolean" ? /> ?
      <wsdl:input>
        <wsoap12:body parts="wsoap12:tParts" ?
                      use="literal|encoded" ?
                      encodingStyle="xs:anyURI" ?
                      namespace="xs:anyURI" ?
                      wsdl:required="xs:boolean" ? />
        <wsoap12:header message="xs:QName"
                        part="xs:NMTOKEN"
                        use="literal|encoded"
                        encodingStyle="xs:anyURI" ?
                        namespace="xs:anyURI" ?
                        wsdl:required="xs:boolean" ? >
          <wsoap12:headerfault message="xs:QName"
                               part="xs:NMTOKEN"
                               use="literal|encoded"
                               encodingStyle="xs:anyURI" ?
                               namespace="xs:anyURI" ?
                               wsdl:required="xs:boolean" ? /> *
        </wsoap12:header> *
      </wsdl:input> ?
      <wsdl:output>
        <!-- Same as wsdl:input -->
      </wsdl:output> ?
      <wsdl:fault>
        <wsoap12:fault name="xs:NMTOKEN"
                       use="literal | encoded"
                       encodingStyle="xs:anyURI"?
                       namespace="xs:anyURI"?
                       wsdl:required="xs:boolean"? />
      </wsdl:fault> *
    </wsdl:operation> *
  </wsdl:binding> *

  <wsdl:service ...>
    <wsdl:port ...>
      <wsoap12:address location="xs:anyURI"
                       wsdl:required="xs:boolean"? />
    </wsdl:port> *
  </wsdl:service> *

</wsdl:definitions>
```

3.1 wsoap12:binding Element

The purpose of the `wsoap12:binding` element is to signify that the binding is bound to the SOAP 1.2 protocol.

```
<wsdl:definitions ... >
    ...
    <wsdl:binding ... >
        <wsoap12:binding transport="xs:anyURI"
                         style="rpc|document" ? ... />
      ...
    </wsdl:binding>
    ...
</wsdl:definitions>
```

The following describes the content model of the wsoap12:binding element.

/wsoap12:binding

This extension element MUST be present as the first child element of a wsdl:binding element that describes a binding to the SOAP 1.2 protocol.

/wsoap12:binding/@style

The value of the style attribute, if present, is a string that specifies the default style for each operation in the containing wsdl:binding element. The style attribute indicates whether the operations within the containing wsdl:binding element are RPC-oriented (messages containing parameters and return values) or document-oriented (message containing document(s)). RPC-oriented operations are marked by style="rpc", document-oriented operations are marked by style="document". If the style attribute is omitted, each of the operations described in the containing wsdl:binding are implicitly interpreted to have a default style of "document".

/wsoap12:binding/@transport

The value of the REQUIRED transport attribute (of type xs:anyURI) indicates which transport of SOAP this binding corresponds to. The URI value "http://schemas.xmlsoap.org/soap/http" corresponds to the HTTP binding. Other URIs may be used here to indicate other transports (such as SMTP, FTP, etc.).

/wsoap12:binding/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

3.2 wsoap12:operation Element

The wsoap12:operation element provides information for the operation as a whole.

```
<wsdl:definitions ... >
    ...
    <wsdl:binding ... >
        ...
        <wsdl:operation ... >
           <wsoap12:operation soapAction="xs:anyURI" ?
                              soapActionRequired="xs:boolean" ?
                              style="rpc|document" ? ... /> ?
           ...
        </wsdl:operation>
        ...
    </wsdl:binding>
    ...
</wsdl:definitions>
```

The following describes the content model of the wsoap12:operation element.

/wsoap12:operation

When bound to HTTP, exactly one wsoap12:operation extension element MUST be present as the first child of the wsdl:operation element. For the SOAP/HTTP protocol binding, this element is required.

/wsoap12:operation/@soapAction

The OPTIONAL `soapAction` attribute (of type xs:anyURI) specifies the value of the action parameter, carried in the application/soap+xml Content-Type header field, for this operation. The value of this attribute MUST be an absolute URI.

/wsoap12:operation/@soapActionRequired

The `soapActionRequired` attribute (of type xs:Boolean), if present, indicates whether or not the value of the `soapAction` attribute must be conveyed in the request message. If the `soapActionRequired` attribute is omitted, its value defaults to 'true'. When the value of `soapActionRequired` is 'true', the `soapAction` attribute MUST be present.

/wsoap12:operation/@style

The value of the `style` attribute, if present, is a string that specifies the style for the operation. The `style` attribute indicates whether the operation is RPC-oriented (messages containing parameters and return values) or document-oriented (message containing document(s)). If the `style` attribute is omitted from the `wsoap12:operation` element, then the operation inherits the style specified, or implied, by the `wsoap12:binding` element in the containing `wsdl:binding` element.

/wsoap12:operation/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

3.3 wsoap12:body Element

The `wsoap12:body` binding extension element provides information on how to bind the different message parts to the `Body` element of the SOAP 1.2 envelope. The `wsoap12:body` element is used in both RPC-oriented and document-oriented messages, but the style of the enclosing operation has important effects on how the `Body` element of the SOAP 1.2 envelope is structured:

- If the binding style is `rpc` each part is a parameter or a return value and appears inside a wrapper element within the body. The wrapper element is assigned a [local name] of the operation name and a [namespace name] of the value of the `namespace` attribute of the `wsoap12:body` element. Each message part, defined with the @type attribute, (part accessor) is carried as a child element of the wrapper element. Part accessor elements are represented by an element with a [local name] of the value of the `name` attribute of the corresponding `wsdl:part` element and a [namespace name] with no value.
- If the operation style is `document` the child elements of the `soap:Body` MUST be elements as defined by the global element declarations identified by the respective QName values of the `element` attributes of the `wsdl:part` element children of the corresponding `wsdl:message` element. The message parts appear as child elements of the soap:Body element and there are no additional wrappers.

```
<wsdl:definitions ... >
    ...
    <wsdl:binding ... >
```

```
        ...
        <wsdl:operation ... >
           <wsdl:input>
               <wsoap12:body parts="wsoap12:tParts" ?
                             namespace="xs:anyURI" ?
                             use="literal|encoded" ?
                             encodingStyle="xs:anyURI" ? ... />
               ...
           </wsdl:input>
           <wsdl:output>
               <wsoap12:body parts="wsoap12:tParts ?
                             namespace="xs:anyURI" ?
                             use="literal|encoded" ?
                             encodingStyle="xs:anyURI" ?  ... />
               ...
           </wsdl:output>
        </wsdl:operation>
        ...
    </wsdl:binding>
    ...
</wsdl:definitions>
```

The following describes the content model of the `wsoap12:body` element.

/wsoap12:body

> Exactly one `wsoap12:body` extension element MUST be present as the first child of each of the `wsdl:input` and `wsdl:output` elements present in each `wsdl:operation` element in a `wsdl:binding` that describes a SOAP 1.2 endpoint binding.

/wsoap12:body/@parts

> The OPTIONAL `parts` attribute (of type wsoap12:tParts, which is a list of xs:NMTOKENs) indicates which message parts are bound to the SOAP 1.2 `Body` element of the message (other message parts may be bound to other portions of the message such as when SOAP is used in conjunction with the multipart/related MIME binding, or when bound as SOAP header blocks). If the `parts` attribute is omitted, then all parts defined by the associated `wsdl:message` are assumed to be included in the SOAP `Body`.

/wsoap12:body/@namespace

> The `namespace` attribute (of type xs:anyURI), if present, defines the namespace to be assigned to the wrapper element for an rpc-style operation. This attribute is ignored if the `style` attribute of either the `wsoap12:binding` element of the containing `wsdl:binding` or of the `wsoap12:operation` element of the containing `wsdl:operation` is either omitted or has a value of “document”. This attribute MUST be present if the value of the `style` attribute of the `wsoap12:binding` element of the containing `wsdl:binding` is “rpc”. The value of the `namespace` attribute, if present, MUST NOT be a relative URI.

/wsoap12:body/@use

The use attribute, if present, indicates whether the message parts are encoded using some encoding rules, or whether the parts define the concrete schema of the message. If the value is "encoded" the message parts are encoded using some encoding rules as specified by the value, actual or implied, of the encodingStyle attribute. If the value is "literal" then the message parts are literally defined by the schema types referenced.

/wsoap12:body/@encodingStyle

The encodingStyle attribute (of type xs:anyURI), if present, identifies the set of encoding rules used to construct the message. This attribute MUST NOT be present unless the style attribute of the wsoap12:binding element of the containing wsdl:binding has a value of “rpc” and the use attribute on the containing wsoap12:body element has a value of "encoded". The value of the encodingStyle attribute, if present, MUST NOT be a relative URI.

/wsoap12:body/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

3.4 wsoap12:fault Element

The wsoap12:fault element specifies the contents of the SOAP 1.2 Fault’s Reason element.

```
<wsdl:definitions ... >
    ...
    <wsdl:binding ... >
        ...
        <wsdl:operation ... >
           ...
           <wsdl:fault ... >*
               <wsoap12:fault name="xs:NMTOKEN"
                              namespace="xs:anyURI" ?
                              use="literal|encoded" ?
                              encodingStyle="xs:anyURI" ? ... />
               ...
           </wsdl:fault>
        </wsdl:operation>
        ...
    </wsdl:binding>
    ...
</wsdl:definitions>
```

The following describes the content model of the wsoap12:fault element.

/wsoap12:fault

The wsoap12:fault extension element MUST be present as the first child of each wsdl:fault element of each wsdl:operation child element of a wsdl:binding that describes a SOAP 1.2 binding.

/wsoap12:fault/@name

The REQUIRED name attribute (of type xs:NMTOKEN) associates the corresponding `wsdl:fault` defined in the `wsdl:portType` for the containing `wsdl:operation`.

/wsoap12:fault/@namespace

The `namespace` attribute (of type xs:anyURI), if present, defines the namespace to be assigned to the wrapper element for the fault. This attribute is ignored if the `style` attribute of either the `wsoap12:binding` element of the containing `wsdl:binding` or of the `wsoap12:operation` element of the containing `wsdl:operation` is either omitted or has a value of “document”. This attribute MUST be present if the value of the `style` attribute of the `wsoap12:binding` element of the containing `wsdl:binding` is “rpc”. The value of the `namespace` attribute, if present, MUST NOT be a relative URI.

/wsoap12:fault/@use

The `use` attribute, if present, indicates whether the message parts are encoded using some encoding rules, or whether the parts define the concrete schema of the message. If the value is "encoded" the message parts are encoded using some encoding rules as specified by the value, actual or implied, of the `encodingStyle` attribute. If the value is "literal" then the message parts are literally defined by the schema types referenced.

/wsoap12:fault/@encodingStyle

The `encodingStyle` attribute (of type xs:anyURI), if present, identifies the set of encoding rules used to construct the fault message. This attribute MUST NOT be present unless the `style` attribute of the `wsoap12:binding` element of the containing `wsdl:binding` has a value of “rpc” and the `use` attribute on the containing `wsoap12:body` element has a value of "encoded". The value of the `encodingStyle` attribute, if present, MUST NOT be a relative URI.

/wsoap12:fault/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

3.5 wsoap12:address Element

The `wsoap12:address` binding extension element is used to give a port an address (a URI) that clients can use to access the service.

```
<wsdl:definitions ... >
    ...
    <wsdl:port ... >
        <wsoap12:address location="xs:anyURI" ... />
        ...
    </wsdl:port>
    ...
</wsdl:definitions>
```

The following describes the content model of the `wsoap12:address` element.

/wsoap12:address

Exactly one `wsoap12:address` extension element MUST be present as the first child of the `wsdl:port` element that is bound to a wsdl:binding that uses the WSDL 1.1 Binding Extension for SOAP 1.1.

/wsoap12:address/@location

The REQUIRED `location` attribute (of type xs:anyURI) is a URI at which the endpoint can be accessed. The value of the `location` attribute MUST NOT be a relative URI. The URI scheme specified MUST correspond to the transport or transfer protocol specified by the `wsoap12:binding/@transport` attribute of the corresponding `wsdl:binding` of the containing `wsdl:port`.

/wsoap12:address/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

3.6 wsoap12:header Element

The `wsoap12:header` element allows a header to be defined that is transmitted within the SOAP 1.2 `Header` element. It is not necessary to exhaustively list all headers that appear in the SOAP Envelope using `wsoap12:header`.

```
<wsdl:definitions ... >
    ...
    <wsdl:binding ... >
        ...
        <wsdl:operation ... >
           ...
           <wsdl:input ... >*
               <wsoap12:header message="xs:QName"
                               part="xs:NMTOKEN"
                               use="literal|encoded"
                               namespace="xs:anyURI" ?
                               encodingStyle="xs:anyURI" ? ... /> *
               ...
           </wsdl:input>
           <wsdl:output ... >*
               <wsoap12:header message="xs:QName"
                               part="xs:NMTOKEN"
                               use="literal|encoded"
                               namespace="xs:anyURI" ?
                               encodingStyle="xs:anyURI" ? ... /> *
               ...
           </wsdl:output>
        </wsdl:operation>
        ...
    </wsdl:binding>
    ...
</wsdl:definitions>
```

The following describes the content model of the `wsoap12:header` element.

/wsoap12:header/@message

The REQUIRED message attribute (of type xs:QName), together with the parts attribute, indicates which message part is to be bound as children of the SOAP 1.2 Header element of the message. The referenced message need not be the same as the message that defines the SOAP Body.

/wsoap12:header/@parts

The REQUIRED parts attribute (of type xs:NMTOKEN), together with the message attribute, indicates which message part is to be bound as a child of the SOAP 1.2 Header element of the message.

/wsoap12:header/@namespace

The namespace attribute (of type xs:anyURI), if present, defines the namespace to be assigned to the header element serialized with use="encoded". The header is constructed in all cases as if the style attribute of the wsoap12:binding element of the containing wsdl:binding had a value of "document". The value of the namespace attribute, if present, MUST NOT be a relative URI.

/wsoap12:header/@use

The use attribute indicates whether the message parts are encoded using some encoding rules, or whether the parts define the concrete schema of the message. If the value is "encoded" the message parts are encoded using some encoding rules as specified by the value, actual or implied, of the encodingStyle attribute. If the value is "literal" then the message parts are literally defined by the schema types referenced.

/wsoap12:header/@encodingStyle

The encodingStyle attribute (of type xs:anyURI), if present, identifies the set of encoding rules used to construct the message. This attribute MUST NOT be present unless the style attribute of the wsoap12:binding element of the containing wsdl:binding has a value of "rpc" and the use attribute on the containing wsoap12:body element has a value of "encoded". The value of the encodingStyle attribute, if present, MUST NOT be a relative URI.

/wsoap12:header/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

/wsoap12:header/wsoap12:headerfault

Optional wsoap12:headerfault elements which appear inside wsoap12:header elements specify the header type(s) that are used to transmit error information pertaining to the header defined by the wsoap12:header.

3.7 wsoap12:headerfault Element

```
<wsdl:definitions ... >
    ...
    <wsdl:binding ... >
        ...
        <wsdl:operation ... >
           ...
           <wsdl:input ... >
```

```
                <wsoap12:header ... >
                    <wsoap12:headerfault message="xs:QName"
                                         part="xs:NMTOKEN"
                                         use="literal|encoded"
                                         namespace="xs:anyURI" ?
                                         encodingStyle="xs:anyURI" ?
                                         ... /> *
                </wsoap12:header> *
                ...
            </wsdl:input> *
            <wsdl:output ... >
                <wsoap12:header ... >
                    <wsoap12:headerfault message="xs:QName"
                                         part="xs:NMTOKEN"
                                         use="literal|encoded"
                                         namespace="xs:anyURI" ?
                                         encodingStyle="xs:anyURI" ?
                                         ... /> *
                </wsoap12:header> *
                ...
            </wsdl:output> *
        </wsdl:operation>
        ...
    </wsdl:binding>
    ...
</wsdl:definitions>
```

The following describes the content model of the wsoap12:headerfault element.

/wsoap12:headerfault/@message

> The REQUIRED message attribute (of type xs:QName), together with the parts attribute, indicates which message part is to be bound as children of the SOAP 1.2 Header element of the message for returning faults pertaining to the enclosing wsoap12:header. The referenced message need not be the same as the message that defines the SOAP Body.

/wsoap12:headerfault/@parts

> The REQUIRED parts attribute (of type xs:NMTOKEN), together with the message attribute, indicates which message part is to be bound as children of the SOAP 1.2 Header element of the message for returning faults pertaining to the enclosing wsoap12:header.

/wsoap12:headerfault/@namespace

> The namespace attribute (of type xs:anyURI), if present, defines the namespace to be assigned to the wrapper element for an rpc-style operation. This attribute is ignored if the style attribute of either the wsoap12:binding element of the containing wsdl:binding or of the wsoap12:operation element of the containing wsdl:operation is either omitted or has a value of “document”. This attribute MUST be present if the value of the style

attribute of the `wsoap12:binding` element of the containing `wsdl:binding` is “rpc”. The value of the `namespace` attribute, if present, MUST NOT be a relative URI.

/wsoap12:headerfault/@use

The `use` attribute indicates whether the message parts are encoded using some encoding rules, or whether the parts define the concrete schema of the message. If the value is "encoded" the message parts are encoded using some encoding rules as specified by the value, actual or implied, of the `encodingStyle` attribute. If the value is "literal" then the message parts are literally defined by the schema types referenced.

/wsoap12:headerfault/@encodingStyle

The `encodingStyle` attribute (of type xs:anyURI), if present, identifies the set of encoding rules used to construct the message. This attribute MUST NOT be present unless the `style` attribute of the `wsoap12:binding` element of the containing `wsdl:binding` has a value of “rpc” and the `use` attribute on the containing `wsoap12:body` element has a value of "encoded". The value of the `encodingStyle` attribute, if present, MUST NOT be a relative URI.

/wsoap12:headerfault/@{any}

This is an extensibility mechanism to allow additional attributes, defined in a foreign namespace, to be added to the element.

4. Security

To avoid breaking signatures, intermediates MUST NOT change the XML representations defined herein. Specifically, intermediaries MUST NOT rewrite XML namespace prefix mappings. Similarly, intermediaries MUST NOT remove XML content that explicitly indicates otherwise-implied content, and intermediaries MUST NOT insert XML content to make implied values explicit. For instance, if a soapActionRequired attribute is present with a value of "false" an intermediary MUST NOT remove it; similarly, if there is no soapActionRequired attribute, an intermediary MUST NOT add one.

5. Acknowledgements

This specification has been developed as a result of joint work with many individuals and teams, including:

Martin Chapman, Oracle

Francisco Curbera, IBM

Alex DeJarnatt, Microsoft

Timm Falter, SAP

Steve Graham, IBM

Ashok Malhotra, Oracle

Noah Mendelsohn, IBM

Greg Pavlik, Oracle

Daniel Roth, Microsoft

Arthur Ryman, IBM

Richard Sitze, IBM

Tony Storey, IBM

Greg Truty, IBM

Sanjiva Weerawarana, IBM

Volker Wiechers, SAP

6. 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 1.2 Part 1]

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/.)

[SOAP 1.2 Part 2]

M. Gudgin, et al, "SOAP Version 1.2 Part 2: Adjuncts," June 2003. (See http://www.w3.org/TR/2003/REC-soap12-part2-20030624/.)

[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.)

[XML Information Set]

John Cowen, et al, "XML Information Set (Second Edition)," February 2004. (See http://www.w3.org/TR/2004/REC-xml-infoset-20040204.)

[XML Schema, Part 1]

H. Thompson, et al, "XML Schema Part 1: Structures," October 2004. (See http://www.w3.org/TR/2004/REC-xmlschema-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-xmlschema-2-20041028/.)

[SOAP Media]

M. Baker, M. Nottingham, "RFC3902: The ‘application/soap+xml’ media type," September 2004. (See http://www.ietf.org/rfc/rfc3902.txt.)

Appendix I – XML Schema

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

```
  http://schemas.xmlsoap.org/wsdl/soap12/wsdl11soap12.xsd
```

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

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--


Copyright 2001 - 2006, International Business Machines Corporation and
Microsoft Corporation
All Rights Reserved


License for WSDL 1.1 Binding Extension for SOAP 1.2 Schema Files


The Authors grant permission to copy and distribute the WSDL 1.1
Binding Extension for SOAP 1.2 Schema Files in any medium without fee
```

```
or royalty as long as this notice and license are distributed with them.
The originals of these files can be located at:

http://schemas.xmlsoap.org/wsdl/soap12/wsdl11soap12.xsd

THESE SCHEMA FILES ARE PROVIDED "AS IS," AND THE AUTHORS MAKE NO
REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, REGARDING THESE
FILES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OR TITLE.  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 THESE FILES.

The name and trademarks of the Authors may NOT be used in any manner,
including advertising or publicity pertaining to these files or any
program or service that uses these files, written prior permission.
Title to copyright in these files will at all times remain with the
Authors.

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



-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
           xmlns:wsoap12="http://schemas.xmlsoap.org/wsdl/soap12/"
           targetNamespace="http://schemas.xmlsoap.org/wsdl/soap12/" >

  <xs:import namespace = "http://schemas.xmlsoap.org/wsdl/" />

  <xs:complexType name="tExtensibilityElementOpenAttrs" >
    <xs:complexContent>
      <xs:extension base="wsdl:tExtensibilityElement" >
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="binding" type="wsoap12:tBinding" />
  <xs:complexType name="tBinding" >
    <xs:complexContent>
      <xs:extension base="wsoap12:tExtensibilityElementOpenAttrs" >
        <xs:attribute name="transport" type="xs:anyURI" use="required"
/>
        <xs:attribute name="style" type="wsoap12:tStyleChoice"
use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="tStyleChoice" >
    <xs:restriction base="xs:string" >
      <xs:enumeration value="rpc" />
      <xs:enumeration value="document" />
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="operation" type="wsoap12:tOperation" />
  <xs:complexType name="tOperation" >
    <xs:complexContent>
      <xs:extension base="wsoap12:tExtensibilityElementOpenAttrs" >
        <xs:attribute name="soapAction" type="xs:anyURI" use="optional"
/>
        <xs:attribute name="soapActionRequired" type="xs:boolean"
use="optional" />
        <xs:attribute name="style" type="wsoap12:tStyleChoice"
use="optional" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:element name="body" type="wsoap12:tBody" />
  <xs:attributeGroup name="tBodyAttributes" >
    <!-- The encodingStyle attribute is now a single URI, instead of a
list of URIs -->
    <xs:attribute name="encodingStyle" type="xs:anyURI" use="optional"
/>
    <xs:attribute name="use" type="wsoap12:useChoice" use="optional" />
    <xs:attribute name="namespace" type="xs:anyURI" use="optional" />
  </xs:attributeGroup>
  <xs:simpleType name="tParts">
    <xs:list itemType="xs:NMTOKEN"/>
  </xs:simpleType>
  <xs:complexType name="tBody" >
    <xs:complexContent>
      <xs:extension base="wsoap12:tExtensibilityElementOpenAttrs" >
        <xs:attribute name="parts" type="wsoap12:tParts" use="optional"
/>
        <xs:attributeGroup ref = "wsoap12:tBodyAttributes" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="useChoice" >
```

```
    <xs:restriction base="xs:string" >
      <xs:enumeration value="literal" />
      <xs:enumeration value="encoded" />
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="fault" type="wsoap12:tFault" />
  <xs:complexType name="tFaultRes" abstract="true" >
    <xs:complexContent>
      <xs:restriction base="wsoap12:tBody" >
          <xs:attribute ref="wsdl:required" use="optional" />
        <xs:attribute name="parts" type="wsoap12:tParts"
use="prohibited" />
            <xs:attributeGroup ref="wsoap12:tBodyAttributes" />
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="tFault" >
    <xs:complexContent>
      <xs:extension base="wsoap12:tFaultRes">
        <xs:attribute name="name" type="xs:NCName" use="required" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>


  <xs:element name="header" type="wsoap12:tHeader" />
  <xs:attributeGroup name="tHeaderAttributes" >
    <xs:attribute name="message" type="xs:QName" use="required" />
    <xs:attribute name="part" type="xs:NMTOKEN" use="required" />
    <xs:attribute name="use" type="wsoap12:useChoice" use="required" />
    <xs:attribute name="encodingStyle" type="xs:anyURI" use="optional"
/>
    <xs:attribute name="namespace" type="xs:anyURI" use="optional" />
  </xs:attributeGroup>
  <xs:complexType name="tHeader" >
    <xs:complexContent>
      <xs:extension base="wsoap12:tExtensibilityElementOpenAttrs" >
        <xs:sequence>
          <xs:element ref="wsoap12:headerfault" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
            <xs:attributeGroup ref="wsoap12:tHeaderAttributes" />
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="headerfault" type="wsoap12:tHeaderFault" />
  <xs:complexType name="tHeaderFault" >
      <xs:attributeGroup ref="wsoap12:tHeaderAttributes" />
      <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>

  <xs:element name="address" type="wsoap12:tAddress" />
  <xs:complexType name="tAddress" >
    <xs:complexContent>
      <xs:extension base="wsoap12:tExtensibilityElementOpenAttrs" >
        <xs:attribute name="location" type="xs:anyURI" use="required"
/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

</xs:schema>
```