



Web Services Description Language (WSDL) Version 1.2 Part 3: Bindings

W3C Working Draft 11 June 2003

This version:

<http://www.w3.org/TR/2003/WD-wsd112-bindings-20030611>

Latest version:

<http://www.w3.org/TR/wsd112-bindings>

Previous versions:

<http://www.w3.org/TR/2003/WD-wsd112-bindings-20030124>

Editors:

Jean-Jacques Moreau, Canon
Jeffrey Schlimmer, Microsoft

This document is also available in these non-normative formats: postscript, PDF, XML, and plain text.

Copyright © 2003 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. WSDL Version 1.2 Bindings describes how to use WSDL in conjunction with SOAP 1.2 [*SOAP 1.2 Part 1: Messaging Framework [p.34]*], HTTP/1.1 GET/POST [*IETF RFC 2616 [p.33]*], and MIME [*IETF RFC 2045 [p.33]*]. This specification depends on WSDL Version 1.2 [*WSDL 1.2 Core Language [p.34]*].

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

This is a W3C Working Draft of the WSDL Version 1.2 Bindings specification for review by W3C members and other interested parties.

This document has been produced as part of the W3C Web Services Activity. The authors of this document are the Web Services Description Working Group members.

This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". This is work in progress and does not imply endorsement by, or the consensus of, either W3C or members of the Web Services Description Working Group.

For a detailed list of changes since the last publication of this document, refer to appendix **B. Part 2 Change Log** [p.35] .

Comments on this document are invited and are to be sent to public-ws-desc-comments@w3.org (public archive). It is inappropriate to send discussion emails to this address. Discussion of this document takes place on the public www-ws-desc@w3.org mailing list (public archive).

Patent disclosures relevant to this specification may be found on the Working Group's patent disclosure page.

A list of current W3C Recommendations and other technical documents can be found at <http://www.w3.org/TR>.

Short Table of Contents

1. Introduction [p.4]
 2. WSDL SOAP Binding [p.6]
 3. WSDL HTTP Binding [p.23]
 4. MIME Binding [p.31]
 5. References [p.33]
 - A. Acknowledgements [p.35] (Non-Normative)
 - B. Part 2 Change Log [p.35] (Non-Normative)
-

Table of Contents

1. Introduction [p.4]
 - 1.1 Notational Conventions [p.5]
2. WSDL SOAP Binding [p.6]
 - 2.1 binding Element with binding [parent] [p.7]
 - 2.1.1 protocol Attribute with binding [owner] [p.7]
 - 2.1.2 styleDefault Attribute with binding [owner] [p.8]
 - 2.1.3 namespaceDefault Attribute with binding [owner] [p.8]
 - 2.1.4 encodingStyleDefault Attribute with binding [owner] [p.8]
 - 2.2 module Element with binding [parent] [p.9]
 - 2.2.1 uri Attribute with module [owner] [p.10]
 - 2.2.2 required Attribute with binding [owner] [p.10]
 - 2.3 propertyConstraint Element with module [parent] [p.10]
 - 2.3.1 uri Attribute with propertyConstraint [owner] [p.11]
 - 2.3.2 type Attribute with propertyConstraint [owner] [p.11]
 - 2.4 operation Element with operation [parent] [p.12]

- 2.4.1 style Attribute with operation [owner] [p.12]
- 2.4.2 soapAction Attribute with operation [owner] [p.13]
- 2.5 body Element with input or output [parent] [p.13]
 - 2.5.1 namespace Attribute with body [owner] [p.14]
 - 2.5.2 encodingStyle Attribute with body [owner] [p.15]
- 2.6 header Element with input or output [parent] [p.15]
 - 2.6.1 element Attribute with header [owner] [p.16]
 - 2.6.2 type Attribute with header [owner] [p.16]
 - 2.6.3 localname Attribute with header [owner] [p.17]
 - 2.6.4 namespace Attribute with header [owner] [p.17]
 - 2.6.5 encodingStyle Attribute with header [owner] [p.17]
 - 2.6.6 role Attribute with header [owner] [p.18]
- 2.7 headerfault Element with input or output [parent] [p.18]
 - 2.7.1 message Attribute with headerfault [owner] [p.19]
 - 2.7.2 part Attribute with headerfault [owner] [p.20]
 - 2.7.3 namespace Attribute with headerfault [owner] [p.20]
 - 2.7.4 encodingStyle Attribute with headerfault [owner] [p.20]
- 2.8 fault Element with fault [parent] [p.21]
 - 2.8.1 message Attribute with fault [owner] [p.22]
 - 2.8.2 namespace Attribute with fault [owner] [p.22]
 - 2.8.3 encodingStyle Attribute with fault [owner] [p.22]
- 2.9 address Element with endpoint [parent] [p.23]
 - 2.9.1 location Attribute with address [owner] [p.23]
- 3. WSDL HTTP Binding [p.23]
 - 3.1 binding Element with binding [parent] [p.24]
 - 3.1.1 verbDefault Attribute with binding [owner] [p.25]
 - 3.2 operation Element with operation [parent] [p.25]
 - 3.2.1 verb Attribute with operation [owner] [p.26]
 - 3.2.2 location Attribute with operation [owner] [p.26]
 - 3.3 Method is GET [p.27]
 - 3.3.1 Case of Input Parts Cited in location Attribute [p.27]
 - 3.3.2 Case of Input Parts NOT Cited in location attribute [p.27]
 - 3.3.2.1 Serializing Parts into HTTP Body request [p.27]
 - 3.3.3 Case of Output Parts [p.28]
 - 3.3.3.1 Serializing parts into HTTP Body response [p.28]
 - 3.4 Method is POST [p.28]
 - 3.4.1 XML Encoding Mechanism [p.28]
 - 3.4.1.1 Case of Input Parts Cited in location Attribute [p.28]
 - 3.4.1.2 Case of Input Parts NOT Cited in location attribute [p.28]
 - 3.4.1.3 Case of Output Parts [p.29]
 - 3.4.2 Multipart Encoding Mechanism [p.29]
 - 3.4.2.1 Case of Input Parts Cited in location Attribute [p.29]
 - 3.4.2.2 Case of Input Parts NOT Cited in location Attribute [p.29]
 - 3.4.2.3 Case of Output Parts [p.29]
 - 3.4.3 application/x-www-form-urlencoded Encoding Mechanism [p.29]
 - 3.4.3.1 Case of Input Parts Cited in location Attribute [p.29]

- 3.4.3.2 Case of Input Parts NOT Cited in location Attribute [p.29]
- 3.4.3.3 Case of Output Parts [p.30]
- 3.5 address Element with endpoint [parent] [p.30]
 - 3.5.1 location Attribute with address [owner] [p.30]
- 4. MIME Binding [p.31]
 - 4.1 How the MIME Binding extends WSDL [p.31]
 - 4.2 mime:content [p.32]
 - 4.3 mime:multipartRelated [p.32]
 - 4.4 soap:body [p.33]
 - 4.5 mime:mimeXml [p.33]
- 5. References [p.33]
 - 5.1 Normative References [p.33]
 - 5.2 Informative References [p.34]

Appendices

- A. Acknowledgements [p.35] (Non-Normative)
 - B. Part 2 Change Log [p.35] (Non-Normative)
 - B.1 WSDL Specification Changes [p.35]
 - B.2 XML Schema Changes [p.38]
-

1. Introduction

The Web Services Description Language WSDL Version 1.2 (WSDL) [*WSDL 1.2 Core Language [p.34]*] defines an XML grammar [*XML 1.0 [p.34]*] for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication. WSDL 1.2 Bindings (this document) defines binding extensions for the following protocols and message formats:

- SOAP Version 1.2 [*SOAP 1.2 Part 1: Messaging Framework [p.34]*] (see **2. WSDL SOAP Binding** [p.6]).
- HTTP/1.1 GET/POST [*IETF RFC 2616 [p.33]*] (see **3. WSDL HTTP Binding** [p.23]).
- MIME [*IETF RFC 2045 [p.33]*] (see **4. MIME Binding** [p.31]).

WSDL 1.2 Primer [*WSDL 1.2 Primer [p.34]*] is a non-normative document intended to provide an easily understandable tutorial on the features of the WSDL Version 1.2 specifications.

The Core Language [*WSDL 1.2 Core Language [p.34]*] of the WSDL 1.2 specification describes the core elements of the WSDL language.

1.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [*IETF RFC 2119 [p.33]*].

This specification uses a number of namespace prefixes throughout; they are listed in Table 1-1 [p.5]. Note that the choice of any namespace prefix is arbitrary and not semantically significant (see [*XML Information Set [p.34]*]).

Table 1-1. Prefixes and Namespaces used in this specification

| Prefix | Namespace | Notes |
|--------|---|---|
| wsdl | "http://www.w3.org/2003/06/wsdl" | A normative XML Schema [<i>XML Schema Structures [p.34]</i>], [<i>XML Schema Datatypes [p.34]</i>] document for the "http://www.w3.org/2003/06/wsdl" namespace can be found at http://www.w3.org/2003/06/wsdl . |
| soap | "http://www.w3.org/2003/06/wsdl/soap12" | A normative XML Schema [<i>XML Schema Structures [p.34]</i>], [<i>XML Schema Datatypes [p.34]</i>] document for the "http://www.w3.org/2003/06/wsdl/soap12" namespace can be found at http://www.w3.org/2003/06/wsdl/soap12 . |
| http | "http://www.w3.org/2003/06/wsdl/http" | A normative XML Schema [<i>XML Schema Structures [p.34]</i>], [<i>XML Schema Datatypes [p.34]</i>] document for the "http://www.w3.org/2003/06/wsdl/http" namespace can be found at http://www.w3.org/2003/06/wsdl/http . |
| mime | "http://www.w3.org/2003/06/wsdl/mime" | A normative XML Schema [<i>XML Schema Structures [p.34]</i>], [<i>XML Schema Datatypes [p.34]</i>] document for the "http://www.w3.org/2003/06/wsdl/mime" namespace can be found at http://www.w3.org/2003/06/wsdl/mime . |
| xsd | "http://www.w3.org/2001/XMLSchema" | Defined in the W3C XML Schema specification [<i>XML Schema Structures [p.34]</i>], [<i>XML Schema Datatypes [p.34]</i>]. |

| | | |
|-----|---|--|
| xsi | "http://www.w3.org/2001/XMLSchema-instance" | Defined in the W3C XML Schema specification [XML Schema Structures [p.34]], [XML Schema Datatypes [p.34]]. |
|-----|---|--|

Namespace names of the general form "http://example.org/..." and "http://example.com/..." represent application or context-dependent URIs [IETF RFC 2396 [p.33]].

This specification uses the Extended Backus-Naur Form (EBNF) as described in XML 1.0 [XML 1.0 [p.34]].

All parts of this specification are normative, with the EXCEPTION of pseudo-schemas, examples, and sections explicitly marked as "Non-Normative". Pseudo-schemas are provided for each component, before the description of this component.

2. WSDL SOAP Binding

| | |
|---|----------|
| Editorial note: JCS | 20030115 |
| The WG is actively redesigning the SOAP/HTTP binding to align with changes in SOAP 1.2. | |

| | |
|--|----------|
| Editorial note: JCS | 20030115 |
| This section is not fully converted to a component model compatible with Part 1. | |

WSDL includes a binding for SOAP 1.2 endpoints, which supports the specification of the following protocol specific information:

- An indication that a binding is bound to the SOAP 1.2 protocol.
- A way of specifying an address for a SOAP endpoint.
- The URI for the SOAPAction HTTP header field for the HTTP binding of SOAP.

| | |
|--|----------|
| Editorial note: JCS | 20030115 |
| The SOAP 1.2 HTTP binding does not define a SOAPAction HTTP header field. This is a known issue. | |

- A list of definitions for header blocks that are transmitted as part of the SOAP Envelope

This binding grammar is not an exhaustive specification since the set of SOAP bindings is evolving. Nothing precludes additional SOAP bindings to be derived from portions of this grammar. For example:

- SOAP bindings that do not employ a URI addressing scheme may substitute another addressing scheme by replacing the `soap:address` element defined in **2.9 address Element with endpoint [parent]** [p.23] .
- SOAP bindings that do not require a `SOAPAction` omit the `soapAction` attribute defined in **2.4 operation Element with operation [parent]** [p.12] .

2.1 binding Element with binding [parent]

```
<definitions >
  <binding >
    <soap:binding protocol="uri"
                  styleDefault="document|rpc"?
                  namespaceDefault="uri"?
                  encodingStyleDefault="uri"? />
  </binding>
</definitions>
```

The mandatory *binding element information item* indicates that messages are SOAP Envelopes [*SOAP 1.2 Part 1: Messaging Framework* [p.34]].

The *binding element information item* has the following Infoset properties:

- A [local name] of binding
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/soap12"
- A [parent] of binding in the namespace named "http://www.w3.org/2003/06/wsdl"
- One or more *attribute information items* amongst its [attributes] as follows:
 - A required `protocol` *attribute information item* as described below.
 - An optional `styleDefault` *attribute information item* as described below.
 - An optional `namespaceDefault` *attribute information item* as described below.
 - An optional `encodingStyleDefault` *attribute information item* as described below.

The *binding element information item* has no [children].

2.1.1 protocol Attribute with binding [owner]

The *protocol attribute information item* indicates which underlying protocol to use for SOAP Envelopes within a *binding element information item*. The *protocol attribute information item* has the following Infoset properties:

- A [local name] of protocol

- A [namespace name] which has no value
- An [owner] of binding in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `protocol attribute information item` is `xsd:anyURI`. The value of the `protocol attribute information item` is a URI that identifies a specific protocol to carry SOAP Envelopes. The URI value "http://www.w3.org/2003/05/soap/bindings/HTTP/" corresponds to the HTTP binding in the SOAP specification.

2.1.2 styleDefault Attribute with binding [owner]

The `styleDefault attribute information item` indicates the default serialization style for all operations contained within the [ancestor] binding. The `styleDefault attribute information item` has the following Infoset properties:

- A [local name] of `styleDefault`
- A [namespace name] which has no value
- An [owner] of binding in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `styleDefault attribute information item` is `xsd:string`. The value of the `styleDefault attribute information item` is either "document" or "rpc"; if the `styleDefault attribute information item` is omitted, the value is "document". See below for more information on the semantics of the `style attribute information item`.

2.1.3 namespaceDefault Attribute with binding [owner]

The `namespaceDefault attribute information item` indicates the default namespace to use for any operation contained within the [ancestor] binding with `style attribute information item` with value "rpc". The `namespaceDefault attribute information item` has the following Infoset properties:

- A [local name] of `namespaceDefault`
- A [namespace name] which has no value
- An [owner] of binding in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `namespaceDefault attribute information item` is `xsd:anyURI`. The value of the `namespaceDefault attribute information item` is the XML namespace to use for the 'wrapper' element for the SOAP Envelope body block(s). See below for more information on the semantics of the `namespace attribute information item`.

2.1.4 encodingStyleDefault Attribute with binding [owner]

The `encodingStyleDefault attribute information item` indicates the default encodingStyle to use for any operation contained within the [ancestor] binding. The `encodingStyleDefault attribute information item` has the following Infoset properties:

- A [local name] of `encodingStyleDefault`
- A [namespace name] which has no value
- An [owner] of `binding` in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `encodingStyleDefault` *attribute information item* is *xsd:anyURI*.

The `encodingStyle` and `encodingStyleDefault` *attribute information items* indicate how the type description for the SOAP Envelope component was derived. This information MAY be exploited by tools that generate data structures from the type description.

The [normalized value] "http://www.w3.org/2003/05/soap-envelope/encoding/none" indicates that the type description was not derived as a function of a well-known encoding. The [normalized value] MUST NOT be "". If no `encodingStyleDefault` is in scope, then the [normalized value] is "http://www.w3.org/2003/05/soap-envelope/encoding/none"

For all [normalized value]s of the `encodingStyle` and `encodingStyleDefault` *attribute information items*, the SOAP Envelope component(s) MUST be exactly as described by the type description; the writer of the SOAP Envelope component(s) MUST it exactly as described by the type description.

2.2 module Element with binding [parent]

```
<definitions >
  <binding >
    <soap:module uri="uri"
                 required="boolean"? >
    </soap:module>
  </binding>
</definitions>
```

The optional `module` *element information item* indicates SOAP module(s) supported by a `binding` *element information item*. A SOAP module[*SOAP 1.2 Part 1: Messaging Framework [p.34]*] realizes zero or more SOAP features implemented as one or more SOAP header blocks.

The `module` *element information item* has the following Infoset properties:

- A [local name] of `module`
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/soap12"
- A [parent] of `binding` in the namespace named "http://www.w3.org/2003/06/wsdl"
- One or more *attribute information items* amongst its [attributes] as follows:
 - A required `uri` *attribute information item* as described below

- An optional *required attribute information item* as described below

2.2.1 uri Attribute with module [owner]

The *uri attribute information item* specifies the URI of the SOAP module for the [owner] module . The *uri attribute information item* has the following Infoset properties:

- A [local name] of *uri*
- A [namespace name] which has no value
- An [owner] of *module* in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *uri attribute information item* is `xsd:anyURI`. The value of the *uri attribute information item* is a URI that identifies a specific SOAP module.

2.2.2 required Attribute with binding [owner]

The *required attribute information item* specifies whether the [owner] module is required by the [parent] binding. The *required attribute information item* has the following Infoset properties:

- A [local name] of *required*
- A [namespace name] which has no value
- An [owner] of *module* in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *required attribute information item* is `xsd:boolean`.

Omitting this *attribute information item* is defined as being semantically equivalent to including it with a value of "false".

2.3 propertyConstraint Element with module [parent]

```
<definitions>
  <binding>
    <soap:module>
      <soap:propertyConstraint uri="uri"
                             type="qname"? />
    </soap:module>
  </binding>
</definitions>
```

The optional *propertyConstraint element information item* indicates SOAP module(s) supported by a *module element information item*.

The *propertyConstraint element information item* has the following Infoset properties:

- A [local name] of `propertyConstraint`
- A [namespace name] of "http://www.w3.org/2003/06/wsd/soap12"
- A [parent] of `module` in the namespace named "http://www.w3.org/2003/06/wsd/"
- One or more *attribute information items* amongst its [attributes] as follows:
 - A required *uri attribute information item* as described below
 - An optional *type attribute information item* as described below

The value of the `propertyConstraint` *element information item* is the value that is required for the SOAP property of the [parent] `module` .

The `propertyConstraint` *element information item* has no [children].

2.3.1 uri Attribute with propertyConstraint [owner]

The *uri attribute information item* specifies the URI of the SOAP property for the [ancestor] `module` . The *uri attribute information item* has the following Infoset properties:

- A [local name] of `uri`
- A [namespace name] which has no value
- An [owner] of `propertyConstraint` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *uri attribute information item* is `xsd:anyURI`. The value of the *uri attribute information item* is a URI that identifies a specific SOAP property for the [ancestor] `module` .

2.3.2 type Attribute with propertyConstraint [owner]

The *type attribute information item* specifies the type of the value of the [owner] `propertyConstraint` . The *type attribute information item* has the following Infoset properties:

- A [local name] of `type`
- A [namespace name] which has no value
- An [owner] of `propertyConstraint` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *type attribute information item* is `xsd:QName`. The value of the *type attribute information item* is XML Qualified Name that identifies the type of the value for the [owner] `propertyConstraint` .

2.4 operation Element with operation [parent]

```
<definitions>
  <binding>
    <operation>
      <soap:operation style="document|rpc"?
                    soapAction="uri"? /> ?
    </operation>
  </binding>
</definitions>
```

The optional *operation element information item* provides binding information for the [parent] operation. The *operation element information item* has the following Infoset properties:

- A [local name] of operation
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/soap12"
- A [parent] of operation in the namespace named "http://www.w3.org/2003/06/wsdl"
- Zero or more *attribute information items* amongst its [attributes] as follows:
 - An optional *style attribute information item* as described below
 - An optional *soapAction attribute information item* as described below

The *operation element information item* has no [children].

If the *operation element information item* has no [attributes], it MAY be omitted.

2.4.1 style Attribute with operation [owner]

The *style attribute information item* indicates the serialization style of the operation owner. The *style attribute information item* has the following Infoset properties:

- A [local name] of style
- A [namespace name] which has no value
- An [owner] of operation in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *style attribute information item* is *xsd:string*.

The value of the *style attribute information item* indicates whether the [owner] operation is serialized according to a document-oriented style (message contains document(s)) or remote-procedure-call (RPC) oriented style (message contains parameter(s) and/or return value(s)). This information may be used to select an appropriate programming model. The value of this *attribute information item* also affects the way in which the Fault of the SOAP message is constructed, as explained in section **2.5 body Element with input or output [parent]** [p.13]. If the *style attribute information item* is not specified, it defaults to the value specified in the [ancestor] *binding element information item* in the namespace named

"http://www.w3.org/2003/06/wsd/soap12".

2.4.2 soapAction Attribute with operation [owner]

| | |
|--|----------|
| Editorial note: JCS | 20030115 |
| The SOAP 1.2 HTTP binding does not define a SOAPAction HTTP header. This is a known issue. | |

The *soapAction attribute information item* specifies the value of the HTTP SOAPAction header for the operation owner. The *soapAction attribute information item* has the following Infoset properties:

- A [local name] of `soapAction`
- A [namespace name] which has no value
- An [owner] of `operation` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *soapAction attribute information item* is *xsd:anyURI*.

The value of the *soapAction attribute information item* is the URI to be included in the HTTP SOAPAction header for the [owner] operation. This URI value should be used directly as the value for the HTTP SOAPAction header; no attempt should be made to make a relative URI value absolute when making the request. For the HTTP protocol binding of SOAP, this value is required (it has no default value). For other SOAP protocol bindings, it **MUST NOT** be specified.

2.5 body Element with input or output [parent]

```
<definitions>
  <binding>
    <operation>
      [ <input> | <output> ]
      <soap:body namespace="uri"?
                encodingStyle="uri"? /> ?
      [ </input> | </output> ]
    </operation>
  </binding>
</definitions>
```

The optional *body element information item* specifies properties of the SOAP Envelope body block(s) for the [parent] operation. The *body element information item* has the following Infoset properties:

- A [local name] of `body`
- A [namespace name] of "http://www.w3.org/2003/06/wsd/soap12"
- A [parent] of `input` or `output` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

- Zero or more *attribute information items* amongst its [attributes] as follows:
 - An optional namespace *attribute information item* as described below
 - An optional `encodingStyle` *attribute information item* as described below

The *body element information item* has no [children].

If the *body element information item* has no [attributes], it MAY be omitted.

The *body element information item* provides information on how to assemble the different message parts inside the `Fault` element of the SOAP Envelope. The *body element information item* is used in both RPC-oriented and document-oriented messages, but the style of the enclosing operation has important effects on how the `Fault` section is structured:

- If the operation 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 named identically to the operation name and its namespace is the value of the namespace attribute. Each message part (parameter) appears under the wrapper, represented by an accessor named identically to the corresponding parameter of the call. Parts are arranged in the same order as the parameters of the call.
- If the operation style is `document` there are no additional wrappers, and the message parts appear directly under the SOAP `Fault` element.

The same mechanisms are used to define the content of the `Fault` and parameter accessor elements.

Each part references a concrete schema definition using either the `element` or `type attribute information item`. In the first case, the element referenced by the part will appear directly under the `Body` SOAP Envelope element (for document style bindings) or under an accessor element named after the message part (in `rpc` style). In the second, the type referenced by the part becomes the schema type of the enclosing element (`Body` for document style or part accessor element for `rpc` style).

2.5.1 namespace Attribute with body [owner]

The *namespace attribute information item* defines the XML namespace to use for the SOAP body block(s) for the [ancestor] operation. The *namespace attribute information item* has the following Infoset properties:

- A [local name] of `namespace`
- A [namespace name] which has no value
- An [owner] of `body` in the namespace named "`http://www.w3.org/2003/06/wsdl/soap12`"

The type of the *namespace attribute information item* is `xsd:anyURI`. The value of the *namespace attribute information item* is the XML namespace to use for the 'wrapper' element for the SOAP Envelope body block(s).

If the value of the *style attribute information item* is "document", then the *namespace attribute information item* with [owner] body is not applicable.

2.5.2 encodingStyle Attribute with body [owner]

The *encodingStyle attribute information item* indicates the encoding style that was used to generate type description for the SOAP body block(s) for the [ancestor] operation .

The *encodingStyle attribute information item* has the following Infoset properties:

- A [local name] of encodingStyle
- A [namespace name] which has no value
- An [owner] of body in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *encodingStyle attribute information item* is *xsd:anyURI*.

The [normalized value] "http://www.w3.org/2003/05/soap-envelope/encoding/none" indicates that the type description has not been derived as a function of a well-known encoding. The [normalized value] MUST NOT be "". If no *encodingStyle* is in scope, then the [normalized value] is the *encodingStyleDefault attribute information item*.

2.6 header Element with input or output [parent]

```
<definitions>
  <binding>
    <operation>
      [ <input> | <output> ]
      <soap:header element="qname"?
                  type="qname"?
                  localname="nmtoken"?
                  namespace="uri"?
                  encodingStyle="uri"?
                  role="uri"? />*
      [ </input> | </output> ]
    </operation>
  </binding>
</definitions>
```

The optional *header element information item* specifies SOAP header block(s) for the [ancestor] operation. The *header element information item* has the following Infoset properties:

- A [local name] of header
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/soap12"
- A [parent] of input or output in the namespace named "http://www.w3.org/2003/06/wsdl"

- One or more *attribute information items* amongst its [attributes] as follows:
 - An optional `element` *attribute information item* as described below
 - An optional `type` *attribute information item* as described below
 - An optional `localname` *attribute information item* as described below
 - An optional `namespace` *attribute information item* as described below
 - An optional `encodingStyle` *attribute information item* as described below
 - An optional `role` *attribute information item* as described below

The header *element information item* has no [children].

The header *element information item* allow headers to be defined that are transmitted as SOAP header blocks. It is not necessary to exhaustively list all header blocks that appear in the SOAP Envelope using header *element information items*. For example, extensions (see Language Extensibility and Binding, [WSDL 1.2 Core Language [p.34]], section 4) to WSDL may imply specific headers should be added to the actual payload, and it is not required to list those headers here.

Exactly one of the `element` and `type` *attribute information items* MUST be specified. The `localname` and `namespace` *attribute information items* MUST be specified if and only if the `type` *attribute information item* is specified.

2.6.1 `element` Attribute with header [owner]

The `element` *attribute information item* refers to an XML Schema element declaration component that defines the SOAP header block for the [owner] header . The `element` *attribute information item* has the following Infoset properties:

- A [local name] of `element`
- A [namespace name] which has no value
- An [owner] of header in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `element` *attribute information item* is `xsd:QName`.

2.6.2 `type` Attribute with header [owner]

The `type` *attribute information item* refers to an XML Schema type description that defines the SOAP header block for the [owner] header . The `type` *attribute information item* has the following Infoset properties:

- A [local name] of `type`

- A [namespace name] which has no value
- An [owner] of header in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *type attribute information item* is *xsd:QName*.

The *namespace attribute information item* is used in the same way as with the *body element information item* (see section **2.5 body Element with input or output [parent]** [p.13]), only *style="document"* is assumed since headers do not contain parameters.

The schema referenced MAY include definitions for the *actor* and *mustUnderstand attribute information item* in the namespace named "http://www.w3.org/2003/05/soap-envelope".

2.6.3 localname Attribute with header [owner]

The *localname attribute information item* defines the localname of the SOAP header block for the [owner] header when it has a *type attribute information item*. The *localname attribute information item* has the following Infoset properties:

- A [local name] of localname
- A [namespace name] which has no value
- An [owner] of header in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *localname attribute information item* is *xsd:NCName*.

2.6.4 namespace Attribute with header [owner]

The *namespace attribute information item* defines the XML namespace of the SOAP header block for the [owner] header when it has a *type attribute information item*. The *namespace attribute information item* has the following Infoset properties:

- A [local name] of namespace
- A [namespace name] which has no value
- An [owner] of header in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the *namespace attribute information item* is *xsd:anyURI*.

2.6.5 encodingStyle Attribute with header [owner]

The *encodingStyle attribute information item* indicates the encoding style

that was used to generate type description for the SOAP header block(s) for the [ancestor] operation .

The `encodingStyle` *attribute information item* has the following Infoset properties:

- A [local name] of `encodingStyle`
- A [namespace name] which has no value
- An [owner] of `header` in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `encodingStyle` *attribute information item* is `xsd:anyURI`.

The [normalized value] "http://www.w3.org/2003/05/soap-envelope/encoding/none" indicates that the type description has not been derived as a function of a well-known encoding. The [normalized value] MUST NOT be "". If no `encodingStyle` is in scope, then the [normalized value] is the `encodingStyleDefault` *attribute information item*.

2.6.6 role Attribute with header [owner]

The optional `role` *attribute information item* identifies the SOAP role to which a SOAP header block is targeted. The `role` *attribute information item* has:

- A [local name] of `role`
- A [namespace name] which has no value
- A [specified] property with a value of "true"
- An [owner] of `header` in the namespace named "http://www.w3.org/2003/06/wsdl/soap12"

The type of the `role` *attribute information item* is `xsd:anyURI`. The value of the `role` *attribute information item* is a URI that names a role that a SOAP node can assume. It SHOULD NOT be a relative URI.

A `role` *attribute information item* that is either absent or has an empty value is equivalent to indicating a value of "http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver", i.e., targeting the SOAP header block to an ultimate SOAP receiver.

2.7 headerfault Element with input or output [parent]

```
<definitions>
  <binding>
    <operation>
      [ <input> | <output> ]
      <soap:headerfault message="qname"
        part="nmtoken"
        namespace="uri"?
        encodingStyle="uri"? />*
      [ </input> | </output> ]
    </operation>
  </binding>
</definitions>
```

| | |
|--|------------|
| Editorial note: JJM | 2003/05/23 |
| This section has not yet been finalized. | |

The optional `headerfault element information item` specifies SOAP fault header block(s) for the [ancestor] operation. The SOAP specification states that errors pertaining to header blocks have to be returned in header blocks, and this mechanism allows specifying the format of such fault header blocks.

The `headerfault element information item` has the following Infoset properties:

- A [local name] of `headerfault`
- A [namespace name] of "http://www.w3.org/2003/06/wsd/soap12"
- A [parent] of `input` or `output` in the namespace named "http://www.w3.org/2003/06/wsd/1"
- One or more *attribute information items* amongst its [attributes] as follows:
 - A mandatory *message attribute information item* as described below.
 - A mandatory *part attribute information item* as described below.
 - An optional *namespace attribute information item* as described below.
 - An optional *encodingStyle attribute information item* as described below.

The `headerfault element information item` has no [children].

The `headerfault element information item` allows fault headers to be defined that are transmitted as SOAP fault header blocks, and that carry information pertaining to the faulty SOAP header block defined by the corresponding `header` .

The *namespace attribute information items* MUST be specified if and only if the *type attribute information item* is specified.

2.7.1 message Attribute with headerfault [owner]

The *message attribute information item* refers to an XML Schema element declaration component that defines the SOAP fault header block for the [owner] `headerfault` . The *message attribute information item* has the following Infoset properties:

- A [local name] of `message`
- A [namespace name] which has no value
- An [owner] of `headerfault` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the message *attribute information item* is *xsd:QName*.

2.7.2 part Attribute with headerfault [owner]

The *part attribute information item* refers to an XML Schema type description that defines the SOAP header block for the [owner] *headerfault*. The *part attribute information item* has the following Infoset properties:

- A [local name] of *part*
- A [namespace name] which has no value
- An [owner] of *headerfault* in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *part attribute information item* is *xsd:QName*.

The namespace *attribute information item* is used in the same way as with the *body element information item* (see section **2.5 body Element with input or output [parent]** [p.13]), only *style="document"* is assumed since headers do not contain parameters.

The schema referenced MAY include definitions for the *role* and *mustUnderstand attribute information item* in the namespace named "http://www.w3.org/2003/05/soap-envelope".

2.7.3 namespace Attribute with headerfault [owner]

The namespace *attribute information item* defines the XML namespace of the SOAP fault header block for the [owner] *headerfault* when it has a *type attribute information item*. The namespace *attribute information item* has the following Infoset properties:

- A [local name] of *namespace*
- A [namespace name] which has no value
- An [owner] of *headerfault* in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the namespace *attribute information item* is *xsd:anyURI*.

2.7.4 encodingStyle Attribute with headerfault [owner]

The *encodingStyle attribute information item* indicates the encoding style that was used to generate type description for the SOAP fault header block(s) for the [ancestor] *operation*.

The *encodingStyle attribute information item* has the following Infoset properties:

- A [local name] of *encodingStyle*
- A [namespace name] which has no value

- An [owner] of `headerfault` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the `encodingStyle` *attribute information item* is `xsd:anyURI`.

The [normalized value] "http://www.w3.org/2003/05/soap-envelope/encoding/none" indicates that the type description has not been derived as a function of a well-known encoding. The [normalized value] **MUST NOT** be "". If no `encodingStyle` is in scope, then the [normalized value] is the `encodingStyleDefault` *attribute information item*.

2.8 fault Element with fault [parent]

```
<definitions>
  <binding>
    <operation>
      <fault>
        <soap:fault message="nmtoken"
                    namespace="uri"?
                    encodingStyle="uri"? />*
      </fault>
    </operation>
  </binding>
</definitions>
```

| | |
|--|------------|
| Editorial note: JJM | 2003/05/23 |
| This section has not yet been finalized. | |

The optional `fault` *element information item* specifies SOAP fault for the [ancestor] operation. It has the following Infoset properties:

- A [local name] of `fault`
- A [namespace name] of "http://www.w3.org/2003/06/wsd/soap12"
- A [parent] of `fault` in the namespace named "http://www.w3.org/2003/06/wsd"
- One or more *attribute information items* amongst its [attributes] as follows:
 - A mandatory `message` *attribute information item* as described below.
 - An optional `namespace` *attribute information item* as described below.
 - An optional `encodingStyle` *attribute information item* as described below.

The `fault` *element information item* has no [children].

2.8.1 message Attribute with fault [owner]

The *message attribute information item* refers to an XML Schema element declaration component that defines the SOAP fault header block for the [owner] `fault`. The *message attribute information item* has the following Infoset properties:

- A [local name] of `message`
- A [namespace name] which has no value
- An [owner] of `fault` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *message attribute information item* is *xsd:QName*.

2.8.2 namespace Attribute with fault [owner]

The *namespace attribute information item* defines the XML namespace of the SOAP fault for the [owner] `fault`. The *namespace attribute information item* has the following Infoset properties:

- A [local name] of `namespace`
- A [namespace name] which has no value
- An [owner] of `fault` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *namespace attribute information item* is *xsd:anyURI*.

2.8.3 encodingStyle Attribute with fault [owner]

The *encodingStyle attribute information item* indicates the encoding style that was used to generate type description for the SOAP fault for the [ancestor] `operation`.

The *encodingStyle attribute information item* has the following Infoset properties:

- A [local name] of `encodingStyle`
- A [namespace name] which has no value
- An [owner] of `fault` in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *encodingStyle attribute information item* is *xsd:anyURI*.

The [normalized value] "http://www.w3.org/2003/05/soap-envelope/encoding/none" indicates that the type description has not been derived as a function of a well-known encoding. The [normalized value] MUST NOT be "". If no `encodingStyle` is in scope, then the [normalized value] is the *encodingStyleDefault attribute information item*.

2.9 address Element with endpoint [parent]

```
<definitions>
  <service>
    <endpoint>
      <soap:address location="uri" />
    </endpoint>
  </service>
</definitions>
```

The mandatory *address element information item* contains the endpoint address of an endpoint . The *address element information item* has the following Infoset properties:

- A [local name] of *address*
- A [namespace name] of "http://www.w3.org/2003/06/wsd/soap12"
- A [parent] of *endpoint* in the namespace named "http://www.w3.org/2003/06/wsd/soap12"
- One *attribute information item* amongst its [attributes] as follows:
 - A required *location attribute information item* as described below

The *address element information item* has no [children].

2.9.1 location Attribute with address [owner]

The mandatory *location attribute information item* identifies the endpoint address of an endpoint . The *location attribute information item* has the following Infoset properties:

- A [local name] of *location*
- A [namespace name] which has no value
- An [owner] of *address* in the namespace named "http://www.w3.org/2003/06/wsd/soap12"

The type of the *location attribute information item* is `xsd:anyURI`. The value of the *location attribute information item* is a URI that processes messages bound by the endpoint . The URI scheme of the value of the *location attribute information item* MUST correspond to the transport specified by the value of the *binding attribute information item* of the endpoint *element information item*.

3. WSDL HTTP Binding

| | |
|--|----------|
| Editorial note: JJM | 20030514 |
| This section is not fully converted to a component model compatible with Part 1. | |

The WSDL HTTP binding described in this section is a binding to HTTP 1.1 [*IETF RFC 2616 [p.33]*]. This binding extends WSDL [*WSDL 1.2 Core Language [p.34]*] by adding *element information items* and *attribute information items* to the Infoset of the WSDL *binding element information item* and *service element information item* (in the "http://www.w3.org/2003/06/wsdl" namespace).

The WSDL HTTP binding supports the specification of the following protocol specific information:

- Binding to the HTTP/1.1 protocol.
- An HTTP method to be used for each operation.
- An address for an HTTP endpoint.
- A relative address for each operation (relative to the base address defined by the port).
- Portions of an input message to be serialized in the path of the request URI, query parameters of the request URI, or the body of the request.
- Serializing an output message in the body of the response.

3.1 binding Element with binding [parent]

```
<definitions>
  <binding>
    <http:binding verbDefault="xsd:string"? />
  </binding>
</definitions>
```

The mandatory *binding element information item* indicates that messages are HTTP messages [*IETF RFC 2616 [p.33]*].

The *binding element information item* has the following Infoset properties:

- A [local name] of binding
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/http"
- A [parent] of binding in the namespace named "http://www.w3.org/2003/06/wsdl"
- Zero or one *attribute information items* amongst its [attributes] as follows:
 - An optional *verbDefault attribute information item* as described below.

The *binding element information item* has no [children].

3.1.1 verbDefault Attribute with binding [owner]

The `verbDefault` *attribute information item* indicates the HTTP method used within a *binding element information item*, for example "GET" or "POST".

The `verbDefault` *attribute information item* has the following Infoset properties:

- A [local name] of `verbDefault`
- A [namespace name] which has no value
- An [owner] of `binding` in the namespace named "http://www.w3.org/2003/06/wsdl/http"

The type of the `verbDefault` *attribute information item* is `xsd:string`. The value of the `verbDefault` *attribute information item* is a string that identifies a specific HTTP method.

3.2 operation Element with operation [parent]

```
<definitions>
  <binding>
    <operation>
      <http:operation location="xsd:anyURI"?
                        verb="xsd:string"? />?
    </operation>
  </binding>
</definitions>
```

The optional `operation` *element information item* provides binding information to the [parent] `operation`.

The `operation` *element information item* has the following Infoset properties:

- A [local name] of `operation`
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/http"
- A [parent] of `operation` in the namespace named "http://www.w3.org/2003/06/wsdl"
- Zero to two *attribute information items* amongst its [attributes] as follows:
 - An optional `location` *attribute information item* as defined below.
 - An optional `verb` *attribute information item* as described below.

The `operation` *element information item* has no [children].

If the `operation` *element information item* has no [attributes], it MAY be omitted.

3.2.1 verb Attribute with operation [owner]

The *verb attribute information item* indicates the HTTP method used by a particular *operation element information item*.

The *verb attribute information item* has the following Infoset properties:

- A [local name] of `verb`
- A [namespace name] which has no value
- An [owner] of `operation` in the namespace named "http://www.w3.org/2003/06/wsd/htp"

The type of the *verb attribute information item* is `xsd:string`.

The value of the *verb attribute information item* is a string that identifies a specific HTTP method to be used with this operation. If there is no *verb attribute information item*, the HTTP method to be used is specified by the *verbDefault attribute information item* with [owner] binding . It is a fatal error if there is no *verb attribute information item* or *verbDefault attribute information item*.

3.2.2 location Attribute with operation [owner]

The *location attribute information item* indicates a relative URI for a particular *operation element information item*. The *location attribute information item* may also indicate which portions of an input message are serialized in the path of the request URI (as opposed to simply being serialized as query parameters of the request URI or in the body of the request).

The *location attribute information item* has the following Infoset properties:

- A [local name] of `location`
- A [namespace name] which has no value
- An [owner] of `operation` in the namespace named "http://www.w3.org/2003/06/wsd/htp"

The type of the *location attribute information item* is `xsd:anyURI`.

The *location attribute information item* specifies a template for the relative portion of the request URI for an operation. This URI is combined with the base URI specified in the *address element information item* to form the full URI for the HTTP request to invoke the operation.

Message parts are serialized according to the rules of:

- Section **3.3 Method is GET** [p.27] , if the HTTP method is "GET"
- Section **3.4 Method is POST** [p.28] , if the HTTP method is "POST"

| | |
|--|------------|
| Editorial note: JJM | 2003/05/23 |
| The editors are soliciting input on whether bindings for HTTP PUT or HTTP DELETE would be appropriate. | |

3.3 Method is GET

The rules in this section **MUST** be followed when the HTTP method is "GET".

3.3.1 Case of Input Parts Cited in `location` Attribute

The `location attribute information item` **MAY** cite input message parts to be serialized in the path of the request URI by enclosing a part name within curly braces (e.g. "{partName}"):

- When constructing the request URI, each pair of curly braces (and enclosed part name) is replaced by the corresponding part value.
- A double curly brace (i.e. "{{" or "}}") **MAY** be used to include a single, literal curly brace in the request URI.

Part values **MUST** be URI escaped.

Parts **MUST NOT** be cited more than once within the `location attribute information item`.

Strings enclosed within single curly braces **MUST** be input message part names; any other strings enclosed within single curly braces are a fatal error.

If cited input message parts are defined using an XML Schema type, they **MUST** derive from `xsd:simpleType`, but **MUST NOT** derive from `xsd:hexBinary`, `xsd:base64Binary`, or any simple list type.

3.3.2 Case of Input Parts NOT Cited in `location` attribute

All input message parts **NOT** cited by the `location attribute information item` are serialized as query parameters appended to the request URI.

3.3.2.1 Serializing Parts into HTTP Body request

- Uncited parts with single values (non-list) are serialized as a single name-value parameter pair. The name of the parameter is the name of the uncited part, and the value of the parameter is the value of the uncited part.
- Uncited parts with list values are serialized as one name-value parameter pair per list value. The name of each parameter is the name of the uncited part, and the value of each parameter is the corresponding value in the list.

Part names and values **MUST** be URI escaped.

If uncited input message parts are defined using an XML Schema type, they **MUST** derive from `xsd:simpleType`, **MUST NOT** derive from `xsd:hexBinary` or `xsd:base64Binary`, but **MAY** derive from a simple list type.

3.3.3 Case of Output Parts

Output message parts are serialized as a single XML document in the body of the HTTP response.

3.3.3.1 Serializing parts into HTTP Body response

The root of the document is an XML element whose local name and XML namespace are the `{name}` and `{target namespace}` of the operation, respectively. The root has one child for each part as follows:

- If the `{component reference}` for an uncited part corresponds to an element declaration, the child is the element instance.
- If the `{component reference}` for an uncited part corresponds to a type definition, the child has a local name, XML namespace, and value corresponding to the `{name}` of the part, the `{target namespace}` of the message, and the value of the part, respectively.

The HTTP `Content-Type` header field **MUST** be the media type "application/xml" or any derived XML type ("*/*+xml").

If there are no message parts, the HTTP `Body` **MUST** be empty, and the HTTP `Content-Length` header field **MUST** have a value of "0" (zero).

3.4 Method is POST

The rules in this section **MUST** be followed when the HTTP method is "POST".

| | |
|---|------------|
| Editorial note: JJM | 2003/05/22 |
| How do we select the encoding mechanism to be used? | |

3.4.1 XML Encoding Mechanism

3.4.1.1 Case of Input Parts Cited in `location` Attribute

All input message parts cited by the `location attribute information item` are serialized according to the rules in **3.3.1 Case of Input Parts Cited in `location` Attribute** [p.27] .

3.4.1.2 Case of Input Parts NOT Cited in `location` attribute

All input message parts NOT cited by the `location attribute information item` are serialized according to the rules in **3.3.3.1 Serializing parts into HTTP Body response** [p.28] .

3.4.1.3 Case of Output Parts

All output message parts are serialized according to the rules in **3.3.3 Case of Output Parts** [p.28] .

3.4.2 Multipart Encoding Mechanism

3.4.2.1 Case of Input Parts Cited in `location` Attribute

All input message parts cited by the `location attribute information item` are serialized according to the rules in **3.3.1 Case of Input Parts Cited in location Attribute** [p.27] .

3.4.2.2 Case of Input Parts NOT Cited in `location` Attribute

All input message parts NOT cited by the `location attribute information item` are serialized as a single MIME Multipart document in the body of the HTTP response. The document has one MIME body part for each uncited message part as follows:

- If the {component reference} for an uncited part corresponds to an element declaration, the content of the MIME body part is the element instance. The media type of the MIME body part **MUST** be "application/xml" or any derived XML type ("*/+xml").
- If the {component reference} for an uncited part corresponds to the simple type definition `xsd:hexBinary`, `xsd:base64Binary` or a type derived from one or the other, the content of the MIME body part is the element instance. The media type of the MIME body part **MUST** be the one specified.
- In all other cases, the content of the MIME body part is an XML document with a local name, XML namespace, and value corresponding to the {name} of the part, the {target namespace} of the message, and the value of the part, respectively. The media type of the MIME body part **MUST** be "application/xml".

3.4.2.3 Case of Output Parts

All output message parts are serialized according to the rules in **3.3.3 Case of Output Parts** [p.28] .

3.4.3 `application/x-www-form-urlencoded` Encoding Mechanism

3.4.3.1 Case of Input Parts Cited in `location` Attribute

All input message parts cited by the `location attribute information item` are serialized according to the rules in **3.3.1 Case of Input Parts Cited in location Attribute** [p.27] .

3.4.3.2 Case of Input Parts NOT Cited in `location` Attribute

All input message parts NOT cited by the `location attribute information item` are serialized as a single URLEncoded form in the body of the HTTP request.

- Uncited input parts **MUST** follow the rules in **3.3.2.1 Serializing Parts into HTTP Body request** [p.27]

The HTTP Content-Type header field **MUST** be the media type "x-www-form-urlencoded".

If there is no uncited message part, the HTTP Body **MUST** be empty, and the HTTP Content-Length header field **MUST** have a value of "0" (zero).

3.4.3.3 Case of Output Parts

All output message parts are serialized according to the rules in **3.3.3 Case of Output Parts** [p.28] .

3.5 address Element with endpoint [parent]

```
<definitions>
  <service>
    <endpoint>
      <http:address location="xsd:anyURI" />
    </endpoint>
  </service>
</definitions>
```

The mandatory *address element information item* contains the endpoint address of an *endpoint* . The *address element information item* has the following Infoset properties:

- A [local name] of *address* .
- A [namespace name] of "http://www.w3.org/2003/06/wsdl/http."
- A [parent] of *endpoint* in the namespace named "http://www.w3.org/2003/06/wsdl."
- One *attribute information item* amongst its [attributes] as follows:
 - A required *location attribute information item* as described below.

The *address element information item* has no [children].

3.5.1 location Attribute with address [owner]

The mandatory *location attribute information item* identifies the endpoint address of an *endpoint* . The *location attribute information item* has the following Infoset properties:

- A [local name] of *location* .
- A [namespace name] which has no value.
- An [owner] of *address* in the namespace named "http://www.w3.org/2003/06/wsdl/http".

The type of the `location attribute information item` is `xsd:anyURI`. The value of the `location attribute information item` is a URI that processes messages bound by the `endpoint`. The URI scheme of the value of the `location attribute information item` MUST correspond to the transport specified by the value of the `binding attribute information item` of the `endpoint element information item`.

4. MIME Binding

| | |
|--|----------|
| Editorial note: JJM | 20030514 |
| This binding has not been revised and may be dropped in a future version of this specification | |

WSDL includes a way to bind abstract types to concrete messages in some MIME format. Bindings for the following MIME types are defined:

- "multipart/related", defined in [IETF RFC 2387 [p.33]].
- "text/xml", defined in [IETF RFC 3023 [p.33]].
- "application/x-www-form-urlencoded", defined in Form content types ([HTML 4.01 [p.33]], section 17.13.4).
- Others (by specifying the MIME type string)

The set of defined MIME types is both large and evolving, so it is not a goal for WSDL to exhaustively define XML grammar for each MIME type. Nothing precludes additional grammar to be added to define additional MIME types as necessary. If a MIME type string is sufficient to describe the content, the `mime` element defined below can be used.

4.1 How the MIME Binding extends WSDL

The MIME Binding extends WSDL with the following extension elements:

```
<mime:content part="nmtoken"? type="string"?/>

<mime:multipartRelated>
  <mime:part> *
    <-- mime element -->
  </mime:part>
</mime:multipartRelated>
<mime:mimeXml part="nmtoken"?/>
```

They are used at the following locations in WSDL:

```
<definitions .... >
  <binding .... >
    <operation .... >
      <input .... >
        <-- mime elements -->
      </input>
    <output .... >
```

```

        <-- mime elements -->
    </output>
</operation>
</binding>
</definitions>

```

MIME elements appear under input and output to specify the MIME format. If multiple appear, they are considered to be alternatives.

4.2 mime:content

To avoid having to define a new element for every MIME format, the `mime:content` element may be used if there is no additional information to convey about the format other than its MIME type string.

```
<mime:content part="nmtoken"? type="string"?/>
```

The `part` attribute is used to specify the name of the message part. If the message has a single part, then the `part` attribute is optional. The `type` attribute contains the MIME type string. A `type` value has two portions, separated by a slash (/), either of which may be a wildcard (*). Not specifying the `type` attribute indicates that all MIME types are acceptable.

If the return format is XML [*XML 1.0 [p.34]*], but the schema is not known ahead of time, the generic mime element can be used indicating "text/xml" [*IETF RFC 3023 [p.33]*]:

```
<mime:content type="text/xml"/>
```

A wildcard (*) can be used to specify a family of mime types, for example all text types.

```
<mime:content type="text/*"/>
```

The following two examples both specify all mime types:

```
<mime:content type="*/*/>
<mime:content/>
```

4.3 mime:multipartRelated

The "multipart/related" MIME type aggregates an arbitrary set of MIME formatted parts into one message using the MIME type "multipart/related". The `mime:multipartRelated` element describes the concrete format of such a message:

```

<mime:multipartRelated>
  <mime:part> *
    <-- mime element -->
  </mime:part>
</mime:multipartRelated>

```

The `mime:part` element describes each part of a "multipart/related" message [*IETF RFC 2387 [p.33]*]. MIME elements appear within `mime:part` to specify the concrete MIME type for the part. If more than one MIME element appears inside a `mime:part`, they are alternatives.

4.4 soap:body

When using the MIME binding with SOAP requests [*SOAP 1.2 Part 1: Messaging Framework [p.34]*], it is legal to use the `soap:body` element as a MIME element. It indicates the content type is "text/xml", and there is an enclosing SOAP Envelope.

4.5 mime:mimeXml

To specify XML payloads that are not SOAP compliant (do not have a SOAP Envelope), but do have a particular schema, the `mime:mimeXml` element may be used to specify that concrete schema. The `part` attribute refers to a message part defining the concrete schema of the root XML element. The `part` attribute MAY be omitted if the message has only a single part. The `part` references a concrete schema using the `element` attribute for simple parts or `type` attribute for composite parts .

```
<mime:mimeXml part="nmtoken"?/>
```

5. References

5.1 Normative References

[HTML 4.01]

HTML 4.01 Specification, D. Raggett, A. Le Hors, I. Jacobs, Editors. World Wide Web Consortium, 24 December 1999. This version of the HTML 4.01 Recommendation is <http://www.w3.org/TR/1999/REC-html401-19991224>. The latest version of HTML 4.01 is available at <http://www.w3.org/TR/html401>.

[IETF RFC 2045]

Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, N. Freed, N. Borenstein, Authors. Internet Engineering Task Force, November 1996. Available at <http://www.ietf.org/rfc/rfc2045.txt>.

[IETF RFC 2119]

Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, Author. Internet Engineering Task Force, June 1999. Available at <http://www.ietf.org/rfc/rfc2119.txt>.

[IETF RFC 2387]

The MIME Multipart/Related Content-type, E. Levinson, Authors. Internet Engineering Task Force, August 1998. Available at <http://www.ietf.org/rfc/rfc2387.txt>.

[IETF RFC 2396]

Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, Authors. Internet Engineering Task Force, August 1998. Available at <http://www.ietf.org/rfc/rfc2396.txt>.

[IETF RFC 2616]

Hypertext Transfer Protocol -- HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Authors. Internet Engineering Task Force, June 1999. Available at <http://www.ietf.org/rfc/rfc2616.txt>.

[IETF RFC 3023]

XML Media Types, M. Murata, S. St. Laurent, D. Kohn, Authors. Internet Engineering Task Force, January 2001. Available at <http://www.ietf.org/rfc/rfc3023.txt>.

[SOAP 1.2 Part 1: Messaging Framework]

SOAP Version 1.2 Part 1: Messaging Framework, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. Frystyk Nielsen, Editors. World Wide Web Consortium, 7 May 2003. This version of the "SOAP Version 1.2 Part 1: Messaging Framework" Specification is <http://www.w3.org/TR/2003/PR-soap12-part1-20030507/>. The latest version of "SOAP Version 1.2 Part 1: Messaging Framework" is available at <http://www.w3.org/TR/soap12-part1/>.

[SOAP 1.2 Part 2: Adjuncts]

SOAP Version 1.2 Part 2: Adjuncts, M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, and H. Frystyk Nielsen, Editors. World Wide Web Consortium, 7 May 2003. This version of the "SOAP Version 1.2 Part 2: Adjuncts" Specification is <http://www.w3.org/TR/2003/PR-soap12-part2-20030507/>. The latest version of "SOAP Version 1.2 Part 2: Adjuncts" is available at <http://www.w3.org/TR/soap12-part2/>.

[XML 1.0]

Extensible Markup Language (XML) 1.0 (Second Edition), T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. World Wide Web Consortium, 10 February 1998, revised 6 October 2000. This version of the XML 1.0 Recommendation is <http://www.w3.org/TR/2000/REC-xml-20001006>. The latest version of XML 1.0 is available at <http://www.w3.org/TR/REC-xml>.

[XML Information Set]

XML Information Set, J. Cowan and R. Tobin, Editors. World Wide Web Consortium, 24 October 2001. This version of the XML Information Set Recommendation is <http://www.w3.org/TR/2001/REC-xml-infoset-20011024>. The latest version of XML Information Set is available at <http://www.w3.org/TR/xml-infoset>.

[XML Schema Structures]

XML Schema Part 1: Structures, H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. World Wide Web Consortium, 2 May 2001. This version of the XML Schema Part 1 Recommendation is <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>. The latest version of XML Schema Part 1 is available at <http://www.w3.org/TR/xmlschema-1>.

[XML Schema Datatypes]

XML Schema Part 2: Datatypes, P. Byron and A. Malhotra, Editors. World Wide Web Consortium, 2 May 2001. This version of the XML Schema Part 2 Recommendation is <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>. The latest version of XML Schema Part 2 is available at <http://www.w3.org/TR/xmlschema-2>.

[WSDL 1.2 Core Language]

Web Services Description Language (WSDL) Version 1.2: Core Language, R. Chinnici, M. Gudgin, J-J. Moreau, S. Weerawarana, Editors. World Wide Web Consortium, 11 June 2003. This version of the "Web Services Description Version 1.2:Core Language" Specification is available at <http://www.w3.org/TR/2003/WD-wsd12-20030611>. The latest version of "Web Services Description Version 1.2: Core Language" is available at <http://www.w3.org/TR/wsd12>.

5.2 Informative References

[WSDL 1.2 Primer]

Web Services Description (WSDL) Version 1.2: Primer, K. Sankar, K. Liu, D. Booth, Editors. World Wide Web Consortium, 11 June 2003. The editors' version of the "Web Services Description Version 1.2: Primer" document is available from <http://www.w3.org/2002/ws/desc/>.

A. Acknowledgements (Non-Normative)

This document is the work of the W3C Web Services Description Working Group.

Members of the Working Group are (at the time of writing, and by alphabetical order): Adi Sakala (IONA Technologies), Alan Davies (SeeBeyond), Allen Brookes (Rogue Wave Software), Amelia Lewis (TIBCO/Extensibility), Arthur Ryman (IBM), Bijan Parsia (University of Maryland), Dale Moberg (Cyclone Commerce), Dan Kulp (IONA Technologies), David Booth (W3C), Dietmar Gaertner (Software AG), Don Mullen (TIBCO Software), Erik Ackerman (Lexmark), Glen Daniels (Macromedia), Igor Sedukhin (Computer Associates), Ingo Melzer (DaimlerChrysler Research and Technology), Jacek Kopecky (Systinet), Jean-Jacques Moreau (Canon), Jeff Mischkin (Oracle Corporation), Jeffrey Schlimmer (Microsoft Corporation), Jerry Thrasher (Lexmark), Jim Hendler (University of Maryland), Johan Pahlsson (L'Échangeur), Jonathan Marsh (*Chair*, Microsoft Corporation), Kevin Canyang Liu (SAP), Laurent De Teneuille (L'Échangeur), Lily Liu (webMethods, Inc.), Martin Gudgin (Microsoft Corporation), Michael Champion (Software AG), Michael Mahan (Nokia), Michael Mealling (Verisign), Mike Ballantyne (Electronic Data Systems), Mike Davoren (W. W. Grainger), Mike McHugh (W. W. Grainger), Philippe Le Hégarret (W3C), Prasad Yendluri (webMethods, Inc.), Roberto Chinnici (Sun Microsystems), Sandeep Kumar (Cisco Systems), Sandra Swearingen (U.S. Department of Defense, U.S. Air Force), Sanjiva Weerawarana (IBM), Steve Graham (Global Grid Forum), Steve Lind (AT&T), Steve Tuecke (Global Grid Forum), Tom Jordahl (Macromedia), Umit Yalcinalp (Oracle Corporation), Waqar Sadiq (Electronic Data Systems), William Vambenepe (Hewlett-Packard Company), Youenn Fablet (Canon)

Previous members were: Aaron Skonnard (DevelopMentor), Don Wright (Lexmark), Barbara Zengler (DaimlerChrysler Research and Technology), Jochen Ruetschlin (DaimlerChrysler Research and Technology), Joyce Yang (Oracle Corporation), Keith Ballinger (Microsoft), Krishna Sankar (Cisco Systems), Mario Jeckle (DaimlerChrysler Research and Technology), Pallavi Malu (Intel Corporation), William Stumbo (Xerox), Daniel Schutzer (Citigroup), Dave Solo (Citigroup), Stefano Pogliani (Sun Microsystems), Stephen White (SeeBeyond), Tim Finin (University of Maryland)

The people who have contributed to discussions on www-ws-desc@w3.org are also gratefully acknowledged.

B. Part 2 Change Log (Non-Normative)

B.1 WSDL Specification Changes

| Date | Author | Description |
|----------|--------|---|
| 20030606 | JJM | Replaced <kw/> by . Indicated that pseudo-schemas are not normative |
| 20030604 | JJM | Reformatted pseudo-syntax elements to match Part 1 layout |
| 20030529 | JCS | Incorporated text to resolve Issue 6e |
| 20030523 | JJM | Commented out MIME binding example; this is primer stuff. |

B.1 WSDL Specification Changes

| | | |
|----------|-----|--|
| 20030523 | JJM | Added pseudo-syntax to all sections. |
| 20030523 | JJM | Started converting the fault and headerfault sections to component model. |
| 20030523 | JJM | Complete the Multipart and x-www-form-urlencoded sections. |
| 20030523 | JJM | Fixed typos in HTTP binding (in particular added NOT in some section headers). |
| 20030522 | JCS | Added rules for serializing HTTP response |
| 20030522 | JCS | Added cardinality to pseudo schema for HTTP binding |
| 20030522 | JCS | Changes @transport to @protocol for SOAP binding |
| 20030522 | JJM | Incorporated remaining text from Philippe into the HTTP binding. |
| 20030522 | JJM | Polished the HTTP binding, split into subsections, added double curly brace escape mechanism, removed pseudo-schema. |
| 20030521 | JCS | Added rules for @verbDefault/@verb and @location. |
| 20030514 | JJM | Start converting the HTTP binding to the component model. The next thing to do will be to remove http:urlReplacement, etc. and incorporate instead Philippe's text. |
| 20030313 | MJG | Changed to Part 3 (from Part 2) |
| 20030117 | JCS | Incorporated resolution for Issue 5 (@encodingStyle). Referenced (rather than in-lined XML Schema). |
| 20030117 | JJM | Various editorial fixes. |
| 20030116 | JCS | Updated pseudo and XML Schema. |
| 20030116 | JJM | Added propertyConstraint section. |
| 20030116 | JJM | Added soap:module section. |
| 20030115 | JCS | Incorporated resolutions for Issue 25 (drop @use and @encoding), Issue 51 (headers reference element/type), and attribute roll up into text and schema. Began reworking SOAP HTTP binding to use Infoset model. Removed informative appendices 'Notes on URIs' and example WSDL documents; expect them to appear in the primer. Updated SOAP 1.2 references to CR. |
| 20030114 | JJM | Removed ednote saying Part 2 is out of synch with Part 1. |
| 20030111 | JJM | Incorporated resolution for issue 17 (role AII). |
| 20030109 | JJM | Incorporated resolution for issue 4 (Namespaces). |
| 20020702 | JJM | Added summary to prefix table. |
| 20020628 | JJM | Added out-of-synch-with-Part2 and not-soap12-yet ednote. |

B.1 WSDL Specification Changes

| | | |
|----------|-----|---|
| 20020621 | JJM | Commented out the link to the previous version. There is no previous version for 1.2 right now. |
| 20020621 | JJM | Rewrote the Notation Conventions section. |
| 20020621 | JJM | Added reference to part 0 in introduction. Renumbered references. |
| 20020621 | JJM | Simplified abstract and introduction. |
| 20020621 | JJM | Obtain the list of WG members from a separate file. |
| 20020621 | JJM | Updated stylesheet and DTDs to latest XMLP stylesheet and DTDs. |
| 20020621 | JJM | Deleted placeholder for appendix C "Location of Extensibility Elements", since this is part 1 stuff and extensibility has been reworked anyway. |
| 20020621 | JJM | Corrected link to issues lists |
| 20020621 | JJM | Updated title from "WSDL" to "Web Services Description Language". Now refer to part 1 as "Web Services... Part 1: Framework" |
| 20020621 | JJM | Added Jeffrey as an editor :-). Removed Gudge (now on Part 2) :-) |
| 20020411 | JJM | Fixed typos noticed by Kevin Liu |
| 20020301 | JJM | Converted the "Schemas" sections |
| 20020301 | JJM | Converted the "Wire WSDL examples" sections |
| 20020301 | JJM | Converted the "Notes on URIs" sections |
| 20020301 | JJM | Converted the "Notational Conventions" sections |
| 20020301 | JJM | Converted the "References" sections |
| 20020301 | JJM | Converted the "MIME Binding" section to XML |
| 20020221 | JJM | Converted the "HTTP Binding" section to XML |
| 20020221 | JJM | Added placeholders for the "Wire examples" and "Schema" sections |
| 20020221 | JJM | Converted the "SOAP Binding" section to XML |
| 20020221 | JJM | Added the Change Log |
| 20020221 | JJM | Added the Status section |
| 20020221 | JJM | Simplified the introduction; referred to Part1 for a longer introduction |
| 20020221 | JJM | Renamed to "Part 2: Bindings" |
| 20020221 | JJM | Created from http://www.w3.org/TR/2001/NOTE-wsdl-20010315 |

B.2 XML Schema Changes

The encoding schema has been updated to be compliant with the XML Schema Recommendations ([*XML Schema Structures* [p.34]] and [*XML Schema Datatypes* [p.34]]). The table below shows the categories of change.

| Class | Meaning |
|-------|---------|
| @ @ @ | @ @ @ |

The table below lists the changes to the encoding schema.

| Class | Description |
|-------|-------------|
| @ @ @ | @ @ @ |