Attachments Profile Version 1.0

Working Group Draft

Date: 2003/12/08 19:39:03

This revision:
http://www.ws-i.org/Profiles/Basic/2003-08/AttachmentsProfile-1.0.htm

Editors:
Chris Ferris, IBM
Anish Karmarkar, Oracle Corp.
Canyang Kevin Liu, SAP AG
Administrative contact:
secretary@ws-i.org

Copyright © 2002-2003 by The Web Services-Interoperability Organization (WS-I) and Certain of its Members. All Rights Reserved.

Abstract

This document defines the WS-I Attachments Profile 1.0, consisting of a set of non-proprietary Web services specifications, along with clarifications and amendments to those specifications that are intended to promote interoperability. This profile compliments the WS-I Basic Profile 1.1 to add support for interoperable SOAP Messages with Attachments-based Web services.

Status of this Document

This document is a Working Group Draft; it has been accepted by the Working Group as reflecting the current state of discussions. It is a work in progress, and should not be considered authoritative or final; other documents may supersede this document.

Feedback

The Web Services-Interoperability Organization (WS-I) would like to receive input, suggestions and other feedback ("Feedback") on this work from a wide variety of industry participants to improve its quality over time.

By sending email, or otherwise communicating with WS-I, you (on behalf of yourself if you are an individual, and your company if you are providing Feedback on behalf of the company) will be deemed to have granted to WS-I, the members of WS-I, and other parties that have access to your Feedback, a non-exclusive, non-transferable, worldwide, perpetual, irrevocable, royalty-free license to use, disclose, copy, license, modify, sublicense or otherwise distribute and exploit in any manner whatsoever the Feedback you provide regarding the work. You acknowledge that you have no expectation of confidentiality with respect to any Feedback you provide. You represent and warrant that you have rights to provide this Feedback, and if you are providing Feedback on behalf of a company, you represent and warrant that you have the rights to provide Feedback on behalf of your company. You also acknowledge that WS-I is not required to review, discuss, use, consider or in any way incorporate your Feedback into future versions of its work. If WS-I does incorporate some or all of your Feedback in a future version of the work, it may, but is not obligated to include your name (or, if you are identified as acting on behalf of your company, the name of your company) on a list of contributors to the work. If the foregoing is not acceptable to you and any company on whose behalf you are acting, please do not provide any Feedback.

Feedback on this document should be directed to wsbasic_comment@ws-i.org.

Table of Contents

1. Introduction

This document defines the WS-I Attachments Profile 1.0 (hereafter, "Profile"), consisting of a set of non-proprietary Web services specifications, along with clarifications to and amplifications of those specifications that are intended to promote interoperability. This profile compliments the WS-I Basic Profile 1.1 to add support for conveying interoperable SOAP Messages with Attachments-based attachments with SOAP messages.

SOAP Messages with Attachments (SwA) defines a MIME multipart/related structure for packaging attachments with SOAP messages. This profile complements the WS-I Basic Profile 1.1 to add support for conveying interoperable SwA-based attachments with SOAP messages.

Section 1 introduces the Profile, and relates the philosophy that it takes with regard to interoperability.

Section 2, "Profile Conformance," explains what it means to be conformant to the Profile.

Each subsequent section addresses a component of the Profile, and consists of two parts; an overview detailing the component specifications and their extensibility points, followed by subsections that address individual parts of the component specifications.

1.1 Relationship to other Profiles

This Profile adds support for SOAP with Attachments and MIME bindings, and is intended to be used in combination with the Basic Profile 1.1.

> **Editors' note:**Old section 2 and 3 have been removed and have to be reconciled with BP 1.1

1.2 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 RFC2119.

Normative statements in the Profile (i.e., those impacting conformance, as outlined in "Profile Conformance") are presented in the following manner:

> Rnnnn*Statement text here.*

where "nnnn" is replaced by the statement number. Each statement contains exactly one requirement level keyword (e.g., "MUST") and one conformance target keyword (e.g., "MESSAGE").

Some statements clarify the referenced specification(s), but do not place additional constraints upon implementations. For convenience, clarifications are annotated in the following manner: c

Some statements are derived from ongoing standardization work on the referenced specification(s). For convenience, such forward-derived statements are annotated in the following manner:

This specification uses a number of namespace prefixes throughout; their associated URIs are listed below. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

- **soap** - "http://schemas.xmlsoap.org/soap/envelope/"
- **xsi** - "http://www.w3.org/2001/XMLSchema-instance"
- **xsd** - "http://www.w3.org/2001/XMLSchema"
- **wsdl** - "http://schemas.xmlsoap.org/wsdl/"
- **soapbind** - "http://schemas.xmlsoap.org/wsdl/soap/"
- **mime** - "http://schemas.xmlsoap.org/wsdl/mime/"
- **wsi** - "http://ws-i.org/schemas/conformanceClaim/"
- **ref** - "http://ws-i.org/profiles/basic/1.1/xsd"

2. Profile Conformance

Editors' note: Conformance Section TBD

3. Attachments

This section of the Profile incorporates the following specifications by reference, and defines extensibility points within them;

- SOAP Messages with Attachments
  Extensibility points:
  - MIME parts - SOAP Messages with Attachments places no restriction on the type of any non-root part in a multipart/related message.

SOAP Messages with Attachments defines a MIME `multipart/related` structure for packaging SOAP envelope with attachments. The Profile mandates the use of that structure, and places the following constraints on its use:

3.1.1 Root Part

R2931 *The entity body of the root part of multipart/related MESSAGE MUST be a soap:Envelope.*

Any MIME part may contain a soap:Envelope, but only the entity body of the root-part of the MIME package is treated as the primary SOAP envelope. Non-root parts are referred to as attachments.

3.1.2 Encoding of Root Part

R2915 *The entity body of the root part of a* `multipart/related` *MESSAGE MUST be* *serialized using either UTF-8 or UTF-16 character encoding.*

R2916 *Non root parts of a `multipart/related` MESSAGE MAY use any character encoding.*

3.1.3 Messages with No Attachments

If a receiver expects zero or more attachments in a message, the sender of that message can use the text/xml media type for a message that has no attachments.

R2917 *A MESSAGE containing zero attachment parts MAY be sent using a content-type of either "text/xml" or "multipart/related" when the WSDL description for the message specifies the mime:multipartRelated element on the corresponding wsdl:input or wsdl:output element in its wsdl:binding.*

This can happen only when the WSDL description specifies a `mime:multipartRelated` that has only one `mime:part` child element containing `soapbind:body`.

For example,

CORRECT:

A WSDL Description that is as follows:

```
<wsdl:definitions
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

targetNamespace="http://example.com/mimewsdl"

xmlns:tns="http://example.com/mimewsdl">

...

    <wsdl:binding name="aBinding" type="tns:aPortType">
        <soapbind:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="anOperation">
            <soap:operation
soapAction="http://example.com/soapaction"/>
            <wsdl:input>
                <mime:multipartRelated>
                    <mime:part>
```

```
                        <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:input>
            <wsdl:output>
                <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
            </wsdl:output>
        </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

may result in an input message which uses the SOAP HTTP Binding as follows:

```
<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">
        <types:anOperation>
 ...
         </types:anOperation>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.4 Dereferencing Attachments

References from mime parts to urls are the same as any other reference using a url, with the same rules applying. The application may do whatever it wishes with the URL resolution. If it chooses to ignore the attachment and resolve the URI from the network, it MAY do so at its peril, but this is its choice.

R2918 *A RECEIVER may ignore a URI reference to an attachment in an envelope.*

3.1.5 Carrying Additional SOAP Envelopes

The Profile places no constraints on the content of attachment parts. Additional xml messages that contain a soap <envelop> may be sent as attachments, but only the root-part of the MIME message should be treated as the primary soap message in a MIME package.

R2919 *A MESSAGE MAY contain `soap:Envelope`s carried as attachments in parts that are not the root part of the message.*

3.1.6 Fault Messages with Attachments

R2920 *An INSTANCE MAY send a fault with attachments if and only if the `wsdl:output`* *element is described using the WSDL MIME binding.*

3.1.7 Ordering of MIME Parts

It is possible that intermediaries might reorder the parts in a `multipart/related` message. Hence semantics should be neither given to, nor implied by the ordering of parts in a message.

R2921 *A RECEIVER MUST NOT infer any semantics from the ordering of non-root MIME parts in a message.*

R2929 *A MESSAGE MAY have its MIME parts in any order provided that the identity of the root part is maintained.*

3.1.8 Position of Root Part

If the `start` parameter is present, then the value of the `start` parameter is the `content-ID` of the root part of the message. In the absence of a `start` parameter, the root part is the first body part in the package, as defined by RFC 2387 Section 3.2.

R2922 *If the `Content-Type` HTTP header field-value in a message does not have a `start` parameter, a RECEIVER MUST treat the first body part of the MIME package as the root part.* c

For example,

CORRECT:

In the message below the first MIME part (which has the Content-ID header of "<rootpart@example.com>") is the root part.

```
MIME-Version: 1.0
Content-Type:  Multipart/Related;  boundary=MIME_boundary;
type=text/xml;
Content-Description: This is the optional message
description.

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@example.com>

<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">
```

```
        <types:SendClaim>

<ClaimDetail>.............................</ClaimDetail
>
            <photo>
              <href>cid:claimphoto@example.com</href>
            </photo>
        </types:SendClaim>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--MIME_boundary
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <claimphoto@example.com>

...binary photograph...
--MIME_boundary--
```

4. MIME Binding

This section of the Profile incorporates the following specifications by reference, and defines extensibility points within them;

- WSDL 1.1, Section 5.0

WSDL 1.1 section 5 defines the MIME binding. The Profile permits the use of the WSDL MIME binding, but limits it to the SOAP Messages with Attachments protocol. The Profile places the following constraints on its use:

4.1.1 Use of MIME Binding Extension

There may be use cases where a sender might be capable of sending a message using SOAP with Attachments yet incapable of receiving and processing such a message.

R2901 *A `wsdl:binding` element in a DESCRIPTION MAY use WSDL MIME Binding as defined in WSDL 1.1 Section 5 on either the wsdl:input or wsdl:output elements, or both.*

R2902 *A SENDER MUST NOT send a message using SOAP with Attachments if the corresponding `wsdl:input` or `wsdl:output` element in the `wsdl:binding` does not specify the WSDL MIME Binding.*

For example,

INCORRECT:

A WSDL Description that is as follows:

```
<wsdl:definitions
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

targetNamespace="http://example.com/mimewsdl"

xmlns:tns="http://example.com/mimewsdl">

...

    <wsdl:binding name="aBinding" type="tns:aPortType">
        <soapbind:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="anOperation">
            <soap:operation
soapAction="http://example.com/soapaction"/>
            <wsdl:input>
                <mime:multipartRelated>
                    <mime:part>
                        <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:input>
            <wsdl:output>
                <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
            </wsdl:output>
        </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

must not result in an output message which uses the MIME Binding as follows:

```
MIME-Version: 1.0
Content-Type:  Multipart/Related;  boundary=MIME_boundary;
type=text/xml;
        start="<rootpart@example.com>"
Content-Description: This is the optional message
description.

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@example.com>

<?xml version='1.0' ?>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">
        <types:anOperationResponse>
...
        </types:anOpertionResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--MIME_boundary--
```

4.1.2 Referencing of Message Parts

A message part in WSDL can be bound to a particular MIME part (using `mime:content`).Unlike a `soapbind:header` which may reference parts contained in a message that is not part of the contract defined by the `wsdl:porttype`, a `mime:content` must not reference a `wsdl:part` that is not defined in the `wsdl:message` referenced by the `wsdl:operation`. Additionally, message parts in WSDL are considered to be an indivisible unit. Components of a message part which is of complex content cannot be selectively bound to a particular MIME part. This does not mean that a part (either bound to the soapbind:Body or to a MIME part) which contains a URI, cannot point to contents carried in a MIME part.

R2903 *A `mime:content` element in a DESCRIPTION MUST NOT reference a `wsdl:part` that is not present in the respective `wsdl:input` or `wsdl:output` of the corresponding `wsdl:operation` of the corresponding `wsdl:portType`..*

R2904 *A `mime:content` element in a DESCRIPTION MUST NOT be bound to a sub-component of an element or type referenced by a `wsdl:part`.*

For example,

INCORRECT:

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types ...>
    <schema xmlns="http://www.w3.org/2001/XMLSchema/"
targetNamespace="http://example.org/foo"
xmlns:ns="http://example.org/foo">
      <element name='foo'>
        <complextContent>
          <sequence>
            <element ref='bar1'/>
            <element ref='bar2'/>
         </sequence>
      </complextContent>
    </element>
  </schema>
</types>
```

```
 ...
<message name='aMsg'>
  <part name='apart' element='ns:foo' />
  <part name="body" element="ns:bar"/>
</message>
<portType>
  <operation>
    <input>
      <part name="apart">
    </input>
...
  </operation>
</portType>
<binding>
  <operation>
    <input>
      <mime:multipartRelated>
        <mime:part>
          <soapbind:body part="body" use="literal"/>
        </mime:part>
        <mime:part>
          <mime:content part="ns:bar1"/>
        </mime:part>
      </mime:multipartRelated>
    </input>
...
  </operation>
</binding>
</definitions>
```

4.1.3 Referencing Attachments from the SOAP Envelope

One of the advantages of having attachments is the ability to include data in a separate MIME part and refer to it from the SOAP envelope. It is important to have the relationship between the referrer and the reference available statically in the WSDL description. Knowledge of this relationship allows tools/platforms to generate artifacts with rich semantics. For example, a generated stub from such a WSDL description can offer a streaming-oriented interface to the attached data.

R2928 *In an ENVELOPE, a URI reference that is typed using the `ref:swaRef` schema type MUST resolve to a MIME part in the same message as the envelope.*

The schema type ref:swaRef is used to indicate that the URI in the instance document points to an attachment in the same message. This type is offered to application/tool/platform developers as an interoperable way to mark references to attachments in the descriptions. Nevertheless, use of other mechanisms does not make the description non-conformant.

The schema for the type used to refer to attachments from the SOAP envelope is:

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

<xsd:schema targetNamespace="http://ws-
i.org/profiles/basic/1.1/xsd"


xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:simpleType name="swaRef">

    <xsd:restriction base="xsd:anyURI" />

  </xsd:simpleType>

</xsd:schema>
```

For example,

CORRECT:

WSDL description for rpc/literal binding:

```
<?xml version="1.0"?>
<wsdl:definitions
xmlns:types="http://example.com/mimetypes"
                  xmlns:ref="http://ws-i.org/bp1.1/xsd"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

targetNamespace="http://example.com/mimewsdl"

xmlns:tns="http://example.com/mimewsdl">

    <wsdl:types>
        <xsd:schema
targetNamespace="http://example.com/mimetypes"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

            <xsd:import namespace="http://ws-
i.org/bp1.1/xsd"
                        schemaLocation="wsi-bp1.1.xsd"
/>

            <xsd:complexType name="swaBinary">
              <xsd:complexContent>
                <xsd:choice minOccurs="1">
                  <xsd:element name="href"
type="ref:swaRef"/>
                  <xsd:element name="binary"
type="xsd:base64Binary"/>
                </xsd:choice>
              </xsd:complexContent>
            </xsd:complexType>
        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="ClaimIn">
        <wsdl:part name="ClaimDetail"
type="xsd:string"/>
        <wsdl:part name="photo"
type="types:swaBinary"/>
    </wsdl:message>
    <wsdl:message name="ClaimOut">
        <wsdl:part name="ClaimRefNo"
type="xsd:string"/>
    </wsdl:message>

    <wsdl:portType name="ClaimPortType">
        <wsdl:operation name="SendClaim">
            <wsdl:input message="ClaimIn"/>
            <wsdl:output message="ClaimOut"/>
        </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="ClaimBinding"
type="tns:ClaimPortType">
        <soapbind:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="SendClaim">
            <soap:operation
soapAction="http://example.com/soapaction"/>
            <wsdl:input>
                <mime:multipartRelated>
                    <mime:part>
                        <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
                    </mime:part>
                    <mime:part>
                        <mime:content part="photo"
type="application/octetstream"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:input>
            <wsdl:output>
                <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
            </wsdl:output>
        </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

Resulting input message for "SendClaim" rpc/literal operation:

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=MIME_boundary;
type=text/xml;
        start="<rootpart@example.com>"
```

```
Content-Description: This is the optional message
description.

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@example.com>

<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">
        <types:SendClaim>

<ClaimDetail>.............................</ClaimDetail
>
            <photo>
              <href>cid:claimphoto@example.com</href>
            </photo>
        </types:SendClaim>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--MIME_boundary
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <claimphoto@example.com>

...binary photograph...
--MIME_boundary--
```

Resulting output message for "SendClaim" rpc/literal operation:

```
<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">
        <types:SendClaimResponse>

<ClaimRefNo>.............................</ClaimRefNo>
        </types:SendClaimResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

CORRECT:

WSDL description for document/literal binding:

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions
xmlns:types="http://example.com/mimetypes"
                  xmlns:ref="http://ws-i.org/bp1.1/xsd"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

targetNamespace="http://example.com/mimewsdl"

xmlns:tns="http://example.com/mimewsdl">

    <wsdl:types>
        <xsd:schema
targetNamespace="http://example.com/mimetypes"

xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">

            <xsd:import namespace="http://ws-
i.org/bp1.1/xsd"
                        schemaLocation="wsi-bp1.1.xsd"
/>

            <xsd:element name="Claim"
type="types:ClaimType"/>
            <xsd:complexType name="ClaimType">
                <xsd:sequence>
                    <xsd:element name="ClaimDetail"
type="xsd:string"/>
                    <xsd:element name="ClaimPhoto"
type="ref:swaRef"/>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:element name="ClaimPhoto"
type="xsd:base64Binary"/>
            <xsd:element name="ClaimRefNo"
type="xsd:string"/>
        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="ClaimIn">
        <wsdl:part name="body" element="types:Claim"/>
        <wsdl:part name="attachment"
element="types:ClaimPhoto"/>
    </wsdl:message>
    <wsdl:message name="ClaimOut">
        <wsdl:part name="out"
element="types:ClaimRefNo"/>
    </wsdl:message>

    <wsdl:portType name="ClaimPortType">
        <wsdl:operation name="SendClaim">
            <wsdl:input message="ClaimIn"/>
            <wsdl:output message="ClaimOut"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ClaimBinding"
type="tns:ClaimPortType">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="SendClaim">
            <soapbind:operation
soapAction="http://example.com/soapaction"/>
            <wsdl:input>
                <mime:multipartRelated>
                    <mime:part>
                        <soapbind:body parts="body"
use="literal"/>
                    </mime:part>
                    <mime:part>
                        <mime:content part="attachment"
type="application/octetstream"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:input>
            <wsdl:output>
                <soapbind:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
</wsdl:definitions>
```

Resulting input message for "SendClaim" document/literal operation:

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=MIME_boundary;
type=text/xml;
        start="<rootpart@example.com>"
Content-Description: This is the optional message
description.

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <rootpart@example.com>

<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">
        <types:Claim>

<ClaimDetail>.............................</ClaimDetail
>

<ClaimPhoto>cid:claimphoto@example.com</ClaimPhoto>
        </types:Claim>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
--MIME_boundary
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <claimphoto@example.com>

...binary photograph...
--MIME_boundary--
```

Resulting output message for "SendClaim" document/literal operation:

```
<?xml version='1.0' ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body
xmlns:types="http://example.com/mimetypes">

<types:ClaimRefNo>...............</types:ClaimRefNo>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Editors' note: The namespace name for schema type ref:swaRef is subject to change until the Profile is final.

4.1.4 Specifying SOAP Headers in Root Part

The WSDL1.1 specification does not specify whether the `soapbind:header` element is permitted as a child of the `mime:part` element along with the `soapbind:body` element. The SOAP Messages with Attachments specification requires that the root part of the multipart message contain the SOAP envelope, but the WSDL1.1 specification is unclear as to how to describe this part. Since the WSDL1.1 specification specifies that the `mime:part` element is used to describe each part of a multipart/related message, the contents of the `mime:part` element that represents the root part of the multipart message must therefore fully describe the SOAP envelope, including the `soapbind:body` and `soapbind:header` elements just as they would be used in the absence of the WSDL MIME binding extension.

R2905 *The `soapbind:header` element in a DESCRIPTION MAY be included as a child of the `mime:part` element.* C

R2906 *A a `soapbind:header` element in a DESCRIPTION MUST NOT be included in a `mime:part` that is not the root part, containing the `soapbind:body` element.* C

For example,

INCORRECT:

```
<wsdl:definitions
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

targetNamespace="http://example.com/mimewsdl"

xmlns:tns="http://example.com/mimewsdl">

...

    <wsdl:binding name="aBinding" type="tns:aPortType">
        <soapbind:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="anOperation">
            <soap:operation
soapAction="http://example.com/soapaction"/>
            <wsdl:input>
                <mime:multipartRelated>
                    <mime:part>
                        <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
                    </mime:part>
                    <mime:part>
                        <soapbind:header
message="tns:headerMessage" part="aPart"
use="literal"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:input>
            <wsdl:output>
 ...
             </wsdl:output>
        </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

4.1.5 MIME Binding Schema Fixes

There are a number of discrepancies between the WSDL1.1 specification and the WSDL MIME binding schema. In the case of the `mime:part` element, the schema incorrectly defines it as a local element declaration and it incorrectly adds a name attribute that is not described in the WSDL1.1 specification.

R2907 *MIME parts in a DESCRIPTION MUST be defined using an element with a local name of `part` in the namespace of the WSDL MIME Binding extension.* C

R2708 *The `mime:part` element in a DESCRIPTION MUST NOT have a `name` attribute.*

These and other fixes to the WSDL MIME Binding extension schema are reflected in the revised schema located at http://schemas.xmlsoap.org/wsdl/mime/xxx.xsd.

4.1.6 Specifying Alternate Media Types

Multiple `mime:content` element children of a `mime:part` are considered alternate acceptable serializations of the referenced `wsdl:part`.

R2909 *Multiple `mime:content` child elements of a `mime:part` element in a DESCRIPTION MUST reference the same `wsdl:part`.*

For example,

INCORRECT:

```
<mime:part>
  <mime:content part="ns:foo" type="image/jpeg"/>
  <mime:content part="ns:bar" type="image/jpeg"/>
</mime:part>
```

CORRECT:

```
<mime:part>
  <mime:content part="ns:foo" type="image/jpeg"/>
  <mime:content part="ns:foo" type="image/gif"/>
</mime:part>
```

4.1.7 WSDL Parts

R2910 *A `mime:part` in a DESCRIPTION MAY reference a `wsdl:part` that is defined using either the `type`attribute or the `element` attribute."*

4.1.8 Specifying Root Part

SOAP Messages with Attachments requires that the root part of the multipart/related package must contain the SOAP envelope, but the WSDL MIME binding is unclear on how this is described.

R2911 *A `mime:multipartRelated` element in a DESCRIPTION MUST contain exactly one `mime:part` element, amongst its child `mime:part` elements, containing a `soapbind:body` child.* C

In a WSDL MIME binding, the `mime:part` that contains a `soapbind:body` describes the root MIME part required by SOAP Messages with Attachments.

For example,

INCORRECT:

```
<wsdl:definitions
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"

targetNamespace="http://example.com/mimewsdl"

xmlns:tns="http://example.com/mimewsdl">

...

    <wsdl:binding name="aBinding" type="tns:aPortType">
        <soapbind:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="anOperation">
            <soap:operation
soapAction="http://example.com/soapaction"/>
            <wsdl:input>
                <mime:multipartRelated>
                    <mime:part>
                        <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
                    </mime:part>
                    <mime:part>
                        <soapbind:body use="literal"
namespace="http://example.com/mimetypes"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:input>
            <wsdl:output>
 ...
             </wsdl:output>
        </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

4.1.9 Ordering of Parts

R2912 *A RECEIVER MUST NOT assume that the order of `mime:part` elements specified in a WSDL description is the same as the order of MIME parts in the message.*

The order of MIME parts specified in a WSDL DESCRIPTION MUST be considered independent of the order of MIME parts in the message.

4.1.10 Sending Fault Messages

R2913 *If the binding for the output message is specified as MIME binding then a Fault MESSAGE MAY* *be sent using SOAP HTTP binding or the MIME binding.*

Editors' note: This requirement will be reworded as the current wording is awkward.

4.1.11 Describing Faults

R2930 *A `wsdl:fault` element in a DESCRIPTION MUST NOT have `mime:multipartRelated` element as its child element.*

4.1.12 Sending Additional Parts Not Described in WSDL

Additional MIME parts may be included in the message beyond those described in the WSDL, and their position or order within the MIME package is not important.

R2923 *A SENDER MAY send non-root MIME parts not described in the WSDL MIME binding.* c

R2926 *A MESSAGE MUST include all of the MIME parts described by its WSDL MIME binding.*

4.1.13 Media Type of Message with Attachments

R2925 *If the WSDL description lists at least one non-root MIME part, the corresponding MESSAGE MUST have a `Content-Type` HTTP header field-value with a media-type of "multipart/related".*

4.1.14 Conformance of SOAP Messages

Profile conformance criteria for the conformance target ENVELOPE do not apply to a non-root part which contains a SOAP envelope. Such a SOAP envelope may be described in the WSDL description as an attachment, in which case, conformance criteria for non-root parts listed in the WSDL description apply.

R2927 *The root part of a MESSAGE MUST be conformant with all the requirements for a MESSAGE in version 1.0 of the Profile.*

Appendix I: Referenced Specifications

The following specifications' requirements are incorporated into the Profile by reference, except where superseded by the Profile:

- SOAP Messages with Attachments
- WSDL 1.1, Section 5.0

Appendix II: Extensibility Points

This section identifies extensibility points, as defined in "Scope of the Profile," for the Profile's component specifications.

These mechanisms are out of the scope of the Profile; their use may affect interoperability, and may require private agreement between the parties to a Web service.

In SOAP Messages with Attachments:

- **MIME parts** - SOAP Messages with Attachments places no restriction on the type of any non-root part in a multipart/related message.

Appendix III: Acknowledgements

This Profile is the work of the WS-I Attachments Profile Working Group, whose members have included:

Mark Allerton (Crystal Decisions Corporation), George Arriola (Talking Blocks, Inc.), Keith Ballinger (Microsoft Corporation), Ilya Beyer (KANA), Rich Bonneau (IONA Technologies), Don Box (Microsoft Corporation), Andrew Brown (Verisign), Heidi Buelow (Quovadx), David Burdett (Commerce One, Inc.), Luis Felipe Cabrera (Microsoft Corporation), Maud Cahuzac (France Telecom), Bhaskar Chakrabarti (United Airlines), Chia Chao (IONA Technologies), Martin Chapman (Oracle Corporation), Richard Chin (Avinon), Roberto Chinnici (Sun Microsystems), Sergio Compean (Avanade, Inc.), Tim Cooke (Onyx Software), Ugo Corda (SeeBeyond Tech), Paul Cotton (Microsoft Corporation), Joseph Curran (Accenture), Ayse Dilber (AT&T), Dave Ehnebuske (IBM), Mark Ericson (Mindreef Inc.), Colleen Evans (Sonic Software), Tim Ewald (Microsoft Corporation), Chuck Fay (FileNet Corporation), Chris Ferris (IBM), Daniel Foody (Actional Corporation), Toru Fujii (NTT), Satoru Fujita (NEC Corporation), Shishir Garg (France Telecom), Yaron Goland (BEA Systems), Hans Granqvist (Verisign), Martin Gudgin (Microsoft Corporation), Marc Hadley (Sun Microsystems), Bob Hall (Unisys Corporation), Scott Hanselman (Corillian), Muir Harding (Autodesk, Inc.), Loren Hart (Verisign), Harry Holstrom (Accenture), Larry Hsiung (Quovadx), Hemant Jain (Tata Consultancy), Steve Jenisch (SAS Institute), Erik Johnson (Epicor Software Corporation), Bill Jones (Oracle Corporation), Menno Jonkers (Tryllian BV), Anish Karmarkar

(Oracle Corporation), Takahiro Kawamura (Toshiba), Bhushan Khanal (WRQ, Inc.), Sunil Kunisetty (Oracle Corporation), Canyang Kevin Liu (SAP AG), Jonathan Marsh (Microsoft Corporation), David Meyer (Plumtree Software, Inc.), Jeff Mischkinsky (Oracle Corporation), Tom Moog (Sarvega Inc.), Gilles Mousseau (Hummingbird Ltd.), Richard Nikula (BMC Software, Inc.), Eisaku Nishiyama (Hitachi, Ltd.), Mark Nottingham (BEA Systems), David Orchard (BEA Systems), Jesse Pangburn (Quovadx), TJ Pannu (ContentGuard, Inc.), Eduardo Pelegri-Llopart (Sun Microsystems), Vijay Rajan (Novell), Eric Rajkovic (Oracle Corporation), Graeme Riddell (Bowstreet), Claus von Riegen (SAP AG), Tom Rutt (Fujitsu Limited), Roger Sanborn (Crystal Decisions Corporation), Krishna Sankar (Cisco Systems, Inc.), Don Schricker (Micro Focus), Dave Seidel (Mindreef Inc.), Akira Shimaya (NTT), Yasser Shohoud (Microsoft Corporation), David Smiley (Mercator Software, Inc.), Seumas Soltysik (IONA Technologies), Joseph Stanko (Plumtree Software, Inc.), Keith Stobie (Microsoft Corporation), Yasuo Takemoto (NTT), Nobuyoshi Tanaka (NEC Corporation), Jorgen Thelin (Cape Clear Software), Sameer Vaidya (Talking Blocks, Inc.), William Vambenepe (Hewlett-Packard), Rick Weil (Eastman Kodak Company), Scott Werden (WRQ, Inc.), Ajamu Wesley (IBM), Shannon Wheatley (Kinzan, Inc.), Ian White (Micro Focus), Sue Worthman (Tryllian BV), Prasad Yendluri (webMethods Inc.).