SOAP 1.1 Binding for MTOM 1.0

March 2, 2006

Authors

Dimitar Angelov, SAP
Christopher Ferris (Editor), IBM
Anish Karmarkar, Oracle
Canyang Kevin Liu, SAP
Jonathan Marsh (Editor), Microsoft
Jeff Mischkinsky, Oracle
Anthony Nadalin, IBM
Ümit Yalçınalp, SAP

Copyright Notice

Abstract

This specification details the necessary modifications to the SOAP Message Transmission Optimization Mechanism (MTOM) and XML-binary Optimized Packaging (XOP) specs necessary to successfully use these technologies with SOAP 1.1.

Status

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

Table of Contents

1. Introduction

This specification details the necessary modifications to the SOAP Message Transmission Optimization Mechanism [MTOM] and XML-binary Optimized Packaging [XOP] specs necessary to successfully use these technologies with [SOAP 1.1].

1.1 Requirements

This specification intends to meet the following requirements:

- Specify the minimal changes to MTOM and XOP to enable these facilities to be used interoperably with SOAP 1.1.
- Allow substantial parts of a SOAP 1.2 MTOM/XOP implementation to be reused with SOAP 1.1.
- Clarify that SOAP 1.1 envelopes so constructed must use XML 1.0 for interoperability.
- Illustrate updates to the [Describing Binary Content] specification (syntax changes) since the publication of XOP.

1.2 Example

Table 1 shows a representation of a SOAP envelope XML Infoset prior to XOP processing. Table 2 shows the same Infoset, serialized using the application/xop+xml format in a MIME Multipart/Related package. These examples correspond to those in [XOP, 1.2 Examples], adjusted to illustrate SOAP 1.1 envelopes.

Table 1: XML Infoset prior to XOP processing (Example 1, SOAP 1.1)

```
(01) <s11:Envelope
(02)     xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
(03)     xmlns:xmime='http://www.w3.org/2005/05/xmlmime'>
(04)   <s11:Body>
(05)     <m:data xmlns:m='http://example.org/stuff'>
(06)       <m:photo
(07)         xmime:contentType='image/png'>/aWKKapGGyQ=</m:photo>
(08)       <m:sig
(09)         xmime:contentType='application/pkcs7-
     signature'>Faa7vROi2VQ=</m:sig>
(10)     </m:data>
(11)   </s11:Body>
(12) </s11:Envelope>
```

Lines (01-02) in Table indicate the message is encoded as SOAP 1.1. Lines (07) and (09) are elements with base64encoded binary data. For purposes of this example, both of these blocks of data will be optimized.

Table 2: Infoset serialized as a XOP package (Example 2, SOAP 1.1)

```
(13) MIME-Version: 1.0
(14) Content-Type: Multipart/Related;boundary=MIME_boundary;
(15)     type="application/xop+xml";
(16)     start="<mymessage.xml@example.org>";
(17)     start-info="text/xml"
(18) Content-Description: A SOAP message with my pic and sig in it
(19) SOAPAction: "http://example.org/action/data"
(20)
(21) --MIME_boundary
(22) Content-Type: application/xop+xml;
(23)     charset=UTF-8;
(24)     type="text/xml"
(25) Content-Transfer-Encoding: 8bit
(26) Content-ID: <mymessage.xml@example.org>
(27)
(28) <s11:Envelope
(29)     xmlns:s11='http://schemas.xmlsoap.org/soap/envelope/'
(30)     xmlns:xmime='http://www.w3.org/2005/05/xmlmime'>
(31)   <s11:Body>
(32)     <m:data xmlns:m='http://example.org/stuff'>
```

```
(33)      <m:photo
(34)  xmime:contentType='image/png'><xop:Include
(35)    xmlns:xop='http://www.w3.org/2004/08/xop/include'
(36)    href='cid:me.png@example.org'/></m:photo>
(37)      <m:sig
(38)  xmime:contentType='application/pkcs7-signature'><xop:Include
(39)    xmlns:xop='http://www.w3.org/2004/08/xop/include'
(40)    href='cid:my.hsh@example.org'/></m:sig>
(41)    </m:data>
(42)  </s11:Body>
(43)</s11:Envelope>
(44)
(45)--MIME_boundary
(46)Content-Type: image/png
(47)Content-Transfer-Encoding: binary
(48)Content-ID: <me.png@example.org>
(49)
(50)// binary octets for png
(51)
(52)--MIME_boundary
(53)Content-Type: application/pkcs7-signature
(54)Content-Transfer-Encoding: binary
(55)Content-ID: <my.hsh@example.org>
(56)
(57)// binary octets for signature
(58)
(59)--MIME_boundary--
(60)
```

Lines (17) and (24) in Table show the media type “text/xml” as required by SOAP 1.1. Lines (28-43) illustrate SOAP 1.1 envelope. Other parts of this package are identical to those one would find in a XOP package for a SOAP 1.2 envelope.

2. Terminology and Notation

2.1 XML Namespaces

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

Table 3: Prefixes and XML Namespaces used in this specification.

Prefix	XML Namespace	Specification(s)
s11	http://schemas.xmlsoap.org/soap/envelope/	[SOAP 1.1]
xmime	http://www.w3.org/2005/05/xmlmime	[Describing Media Content]

2.2 Notational Conventions

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

2.3 Compliance

Normative text within this specification takes precedence over outlines, which in turn take precedence over examples.

3. Using MTOM/XOP with SOAP 1.1

Use of MTOM/XOP with SOAP 1.1 is straightforward. There are differences between SOAP 1.1 and SOAP 1.2 particularly with respect to definitions of SOAP 1.2 Features, SOAP 1.2 Modules, SOAP 1.2 Message Exchange Patterns, SOAP 1.2 Property Conventions for Message Exchange Patterns, SOAP 1.2 HTTP binding description and intermediaries. Those parts of MTOM specification that are specific to SOAP 1.2-only constructs are not applicable to SOAP 1.1 and thus not applicable to this specification. For example, in [MTOM] 2 Abstract SOAP Transmission Optimization Feature and [MTOM] 4 HTTP SOAP Transmission Optimization Feature the feature definition and its effects on SOAP MEP and SOAP MEP properties are not applicable to this specification.

SOAP 1.1 is defined in terms of XML elements, and MTOM describes SOAP 1.2 constructs in terms of information items. There is a clear correspondence between the two, as described in the [XML Information Set].

All constraints described in [MTOM] and [XOP] MUST be followed, except as noted above or changed as specified below.

3.1 Serialization of a SOAP 1.1 message

When sending a SOAP 1.1 message using the MIME Multipart/Related Serialization, the SOAP envelope Infoset is serialized into XML 1.0 as specified in [XOP] 3.1 Creating XOP packages. Specifically:

- The content-type of the outer package MUST be `multipart/related`.
- The type parameter of the content-type header of the outer package MUST have a value of `application/xop+xml` (see [XOP], 4.1 MIME Multipart/Related XOP Packages).
- The start-info parameter of the content-type header of the outer package MUST specify a content-type for the root part of `text/xml`.
- The content-type of the root part MUST be `application/xop+xml` (see [XOP], 4.1 MIME Multipart/Related XOP Packages).
- The type parameter of the content-type header of the root part MUST specify a content-type of `text/xml`.

The result is a MIME Multipart/Related XOP package (see [XOP]): one body part, the root, containing an XML 1.0 representation of the modified SOAP 1.1 envelope, with an additional part used to contain the binary representation of each element that was optimized.

This section corresponds to, and supercedes, [MTOM] 3.2 Serialization of a SOAP message which describes similar treatment for SOAP 1.2.

3.2 Sending a SOAP 1.1 message over HTTP

3.2.1 xop:Include restrictions

Implementations supporting the HTTP SOAP Transmission Optimization binding for SOAP 1.1 MUST enforce the restriction that XOP is not to be used with Infosets that contain *element information items* of name xop:Include (see [XOP], 3. XOP Infosets Constructs]). In any case where a SOAP 1.1 envelope containing such an *element information item* is to be sent, the binding MUST do one of the following:

- Fall back to use the `text/xml` media type or any other suitable media type, i.e., send the SOAP envelope without using the HTTP SOAP Transmission Optimization Feature.
- Generate a binding-dependent SOAP fault.

This section corresponds to and supercedes [MTOM, 4.3.1 Sending a SOAP message] which describes similar treatment for SOAP 1.2.

3.2.2 SOAPAction

An HTTP envelope containing a XOP package MUST specify a SOAPAction header (see [SOAP 1.1], 6.1.1 The SOAPAction HTTP Header Field). However, the value of such header MAY be the empty string: "".

4. Security Considerations

Because SOAP can carry application defined data whose semantics is independent from that of any MIME wrapper (or context within which the MIME wrapper is used), one should not expect to be able to understand the semantics of the SOAP message based on the semantics of the MIME wrapper alone. Therefore, whenever using the `application/xop+xml` media type, it is strongly advised that the security implications of the context within which the SOAP message is used is fully understood. The security implications are likely to involve both the specific SOAP binding to an underlying protocol as well as the application-defined semantics of the data carried in the SOAP message.

It is assumed that such mechanisms that protect SOAP messages at the infoset level will seamlessly adapt to provide protection for messages conforming to this document. It is strongly recommended that the messages be secured using those mechanisms. In order to properly secure messages, the body and all relevant headers need to be included in the signature. It should be noted that for messages traveling through intermediaries, it is possible that some or all of the message information headers may have multiple signatures when the message arrives at the ultimate receiver. It is strongly recommended that the initial sender include a signature to prevent any spoofing by intermediaries.

5. Acknowledgements

Martin Chapman, Oracle

Timm Falter, SAP

Steve Graham, IBM

Maryann Hondo, IBM

Ashok Malhotra, Oracle

Noah Mendelsohn, IBM

Greg Pavlik, Oracle
Jeffrey Schlimmer, Microsoft
Greg Truty, IBM
Volker Wiechers, SAP

6. References

[Describing Media Content]

Anish Karmarkar, et al, "Describing Media Content of Binary Data in XML," May 2005. (See http://www.w3.org/TR/xml-media-types/.)

[MTOM]

Martin Gudgin, et al, "SOAP Message Transmission Optimization Mechanism," January 2005. (See http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/.)

[RFC 2119]

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

[SOAP 1.1]

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

[SOAP 1.2]

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

[XML Information Set]

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

[XOP]

Martin Gudgin, et al, "XML-binary Optimized Packaging," January 2005. (See http://www.w3.org/TR/2005/REC-xop10-20050125/.)