Internet-Draft
Mark Baker
Planetfred, Inc.
Mark Nottingham
January 14, 2002

The "application/soap+xml" media type
draft-baker-soap-media-reg-00.txt

Abstract

This document defines the "application/soap+xml" media type which can be used to describe SOAP 1.2 messages serialized as XML.

1. Introduction

SOAP 1.2 is an XML Infoset[INFOSET] based protocol at the core of which is an envelope that defines a framework for describing what is in a message and how to process it, and a binding framework for exchanging messages using an underlying protocol.

By being based on the XML Infoset, and not XML 1.0 [XML] itself, SOAP permits alternate serializations of messages. The "application/soap+xml" media type can be used to describe those SOAP messages produced with the XML 1.0 serialization.

Feedback or discussion about this draft should be directed to the XML Protocol Working Group public mailing list, xml-dist-app@w3.org with archives at http://lists.w3.org/Archives/Public/xml-dist-app/.

2. Registration

MIME media type name: application
MIME subtype name: soap+xml
Required parameters: none
Optional parameters:
This parameter has identical semantics to the charset parameter of the "application/xml" media type as specified in [XMLMIME].

See Section 6 of this document.

Identical to those of "application/xml" as described in [XMLMIME], Section 3.2.

See Section 3 of this document.

See Section 4 of this document.

See [SOAP12P1] and [SOAP12P2].

No known applications currently use this media type.

There is no single initial byte sequence that is always present for SOAP messages. Section 5 below gives some context for why recognizing a SOAP message without any metadata is problematic, and some guidelines on how the XML 1.0 serialization of the SOAP envelope may be recognized.

SOAP messages are not required or expected to be stored as files.

Macintosh File Type code: TEXT

Mark Baker <mbaker@planetfred.com>

The SOAP 1.2 specification set is a work product of the World Wide Web Consortium’s XML Protocol Working Group. The W3C has change control over these specifications.

3. Security considerations

3.1 Traditional considerations

An obvious target for security considerations is the SOAP processing model. It accommodates intermediaries, so is potentially prone to typical intermediary style attacks. SOAP itself provides no specific protection from these attacks, but instead leaves that to SOAP extensions, or to the underlying protocol(s) to which it is bound.

For example, portions of a SOAP envelope could be encrypted with XML Encryption [XMLENC], protecting the chain from some "man in the middle" attacks.
The SOAP envelope includes two features that form the basis of the intermediary model. Those features are:

- "actor", used to target headers to SOAP intermediaries that understand its URI value
- "mustUnderstand", used to require that a SOAP header must be understood by a SOAP intermediary, or the message rejected

"actor" values are URIs, however there are no requirements that a SOAP processor attempt to resolve them, so no security issues should result from their use.

"mustUnderstand" has no known security issues, outside the generic concerns about intermediaries, that enforcing the end to end model requires extensions, or using features provided by the underlying protocol.

3.2 Considerations of the different uses of SOAP

From a security perspective, SOAP can be seen to be used in two different ways; tunneled, and non-tunneled.

3.2.1 Non-tunneled

The non-tunneled use of SOAP is as an XML Infoset based envelope whose hop-by-hop transfer semantics are inherited from the application protocol used for that hop.

In this use, the binding of SOAP to that protocol extends the application semantics of the protocol without modifying or otherwise disrupting the existing semantics. This includes the safety provided by the fixed interface of the application protocol. For example, when bound to HTTP using the SOAP 1.2 default HTTP binding, the transfer of a SOAP envelope is performed with HTTP POST semantics, and the safety implications are the same as for any other HTTP POST.

3.2 Tunneled

Another use of SOAP is as a framework for building and deploying new protocols, tunneled over the underlying protocol to which it is bound. If the underlying protocol is an application protocol, then any safety afforded by the fixed interface of that protocol would be disregarded (by definition of "tunnel"). It will be up to the designer of the new protocol to ensure that its semantics are safe.

That SOAP explicitly supports tunneling would at first glance appear to be a problem. However, as tunneling over application protocols is already fairly common (including IPP [IPP], on the IETF Internet standards track), the possibility of consolidating future tunneling practice within a framework such as SOAP should help security in the long run.

As a worst case from a security perspective, if SOAP were used only for tunneling, it would be no worse than the tunneling that exists today.

4. Interoperability considerations

There are several factors affecting a SOAP processor’s ability to successfully process a SOAP message. These are itemized in Section
4.4.5 of [SOAP12P1];

- version mismatch; the SOAP envelope uses an unrecognized namespace URI
- mustUnderstand; the SOAP envelope contained a header tagged as "mustUnderstand", but the recipient processor didn’t understand or could not obey them
- DTD not supported; the SOAP message contained a Document Type Definition, which SOAP 1.2 does not support
- Data encoding unknown; a header or body targeted at the current SOAP node is scoped with a data encoding that the current node does not support

SOAP defines faults for these, and requires that conformant processors send the appropriate fault when encountered.

5. Recognizing SOAP messages

SOAP 1.2 does not require or assume that SOAP 1.2 messages have any particular serialization, making it impossible to determine (in the absence of other information) when a chunk of data is a SOAP 1.2 message or not.

However, for the case of the XML 1.0 serialization of SOAP 1.2 messages, the following best describes how these messages may be recognized.

The root element of the message will always be named "envelope" with the namespace "http://www.w3.org/2001/12/soap-envelope". However, this may be expressed in two ways;

- with an xmlns declaration on the root element
- with a prefix-adorned xmlns declaration on the root element where "envelope" is so prefixed

6. The "action" parameter

SOAP 1.1 [SOAP11] introduced the HTTP SOAPAction header, which was designed to be used to indicate the "intent" of the SOAP message. Its value is a URI reference, and its existence was to be used to unambiguously identify SOAP messages transferred with HTTP. It was required because SOAP 1.1 used the generic "text/xml" media type, preventing that same information from being communicated on the media type where it might normally reside.

As SOAP 1.2 is defining its own media type, the possibility of using a parameter on that media type to convey the same information as is done with SOAPAction, now exists. "action" is that parameter.

Its semantics are identical to that of the SOAPAction header of SOAP 1.2, not SOAP 1.1 (a key difference being that it is optional in SOAP 1.2). The following is a mapping of the meanings of the various uses of SOAPAction (as described in [SOAP12P2]) to the "action" parameter;

<table>
<thead>
<tr>
<th>SOAPAction use</th>
<th>&quot;action&quot; parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;<a href="http://foo/bar">http://foo/bar</a>&quot;</td>
<td>action=&quot;<a href="http://foo/bar">http://foo/bar</a>&quot;</td>
</tr>
<tr>
<td>&quot;myapp.sdl&quot;</td>
<td>action=&quot;myapp.sdl&quot;</td>
</tr>
</tbody>
</table>
7. Fragment identifiers

No meaning is associated with fragment identifiers for content described by the "application/soap+xml" media type.

8. Authors’ Addresses

Mark A. Baker
Planetfred, Inc.
44 Byward Market, Suite 240
Ottawa, Ontario, CANADA. K1N 7A2
tel:+1-613-789-1818
mailto:mbaker@planetfred.com

Mark Nottingham
mailto:mnot@mnot.net

9. References


