Network Working Group
Request for Comments: 3288
Category: Standards Track

E. O'Tuathail
Clipcode.com
M. Rose
Dover Beach Consulting, Inc.

June 2002

Using the Simple Object Access Protocol (SOAP) in Blocks Extensible Exchange Protocol (BEEP)

Status of this Memo

Copyright Notice

Abstract

This memo specifies a Simple Object Access Protocol (SOAP) binding to the Blocks Extensible Exchange Protocol core (BEEP). A SOAP binding describes how SOAP messages are transmitted in the network.

The SOAP is an XML-based (extensible markup language) messaging protocol used to implement a wide variety of distributed messaging models. It defines a message format and describes a variety of message patterns, including, but not limited to, RPC, asynchronous event notification, unacknowledged messages, and forwarding via SOAP intermediaries.

Table of Contents

1. Introduction

This memo specifies how SOAP 1.1 envelopes[1] are transmitted using a BEEP profile[2]. In the W3C, the XMLP effort is evolving SOAP. Accordingly, this memo provides a mechanism for negotiating the use of new features.

Throughout this memo, the term "envelope" refers to the "SOAP-Env:Envelope" element defined in Section 4 of [1]. Further, the terms "peer", "client", "server", "one-to-one", and "one-to-many" are used in the context of BEEP. In particular, Sections 2.1 and 2.1.1 of [2] discuss BEEP roles and exchange styles.

2. BEEP Profile Identification

The BEEP profile for SOAP is identified as

    http://iana.org/beep/soap

in the BEEP "profile" element during channel creation.

In BEEP, when the first channel is successfully created, the "serverName" attribute in the "start" element identifies the "virtual host" associated with the peer acting in the server role, e.g.,

```
<start number='1' serverName='stockquoteserver.example.com'>
    <profile uri='http://iana.org/beep/soap' />
</start>
```

The "serverName" attribute is analagous to HTTP's "Host" request-header field (c.f., Section 14.23 of [3]).

There are two states in the BEEP profile for SOAP, "boot" and "ready":

- In the "boot" state, the peer requesting the creation of the channel sends a "bootmsg" (either during channel initialization or in a "MSG" message).
  - If the other peer sends a "bootrpy" (either during channel initialization or in a "RPY" message), then the "ready" state is entered
  - Otherwise, the other peer sends an "error" (either during channel initialization or in a "ERR" message), then no state change occurs.
- In the "ready" state, either peer begins a SOAP message pattern by sending a "MSG" message containing an envelope. The other peer completes the message pattern either by:
  - sending back a "RPY" message containing an envelope; or,
  - sending back zero or more "ANS" messages, each containing an envelope, followed by a "NUL" message.

  Regardless, no state change occurs.

2.1 Profile Initialization

The boot message is used for two purposes:

resource identification: each channel bound to the BEEP profile for SOAP provides access to a single resource (a network data object or service).

feature negotiation: if new features of SOAP (such as compression) emerge, their use can be negotiated.

The DTD syntax for the boot message and its response are:

```
<!ELEMENT bootmsg     EMPTY>
<!ATTLIST bootmsg
          resource    CDATA             #REQUIRED
          features    NMTOKENS          "">

<!ELEMENT bootrpy     EMPTY>
<!ATTLIST bootrpy
          features    NMTOKENS          "">
```

The boot message contains a mandatory and an optional attribute:

- the "resource" attribute, which is analagous to HTTP's "abs_path" Request-URI parameter (c.f., Section 5.1.2 of [3]); and,

- the "features" attribute, which, if present, contains one or more feature tokens, each indicating an optional feature of the BEEP profile for SOAP that is being requested for possible use over the channel.

Section 6.1 defines a registration template for optional features.

If the peer acting in the server role recognizes the requested resource, it replies with the boot response that contains one optional attribute:

- the "features" attribute, if present, contains a subset of the feature tokens in the boot message, indicating which features may be used over the channel. (If not present or empty, then no features may be used.)

Otherwise, if the boot message is improperly formed, or if the requested resource isn't recognized, the peer acting in the server role replies with an error message (c.f., Section 7.1 of [2]).

Typically, the boot message and its response are exchanged during channel initialization (c.f., Section 2.3.1.2 of [2]).

For example, here the boot message and its response are exchanged during channel initialization:

```
C: <start number='1' serverName='stockquoteserver.example.com'>
C:     <profile uri='http://iana.org/beep/soap'>
C:         <![CDATA[<bootmsg resource='/StockQuote' />]]>
C:     </profile>
C: </start>

S: <profile uri='http://iana.org/beep/soap'>
S:     <![CDATA[<bootrpy />]]>
S: </profile>
```

The channel bound to the BEEP profile for SOAP is now in the "ready" state.

Alternatively, here is an example in which the boot exchange is unsuccessful:

```
C: <start number='1' serverName='stockquoteserver.example.com'>
C:     <profile uri='http://iana.org/beep/soap'>
C:         <![CDATA[<bootmsg resource='/StockPick' />]]>
C:     </profile>
C: </start>

S: <profile uri='http://iana.org/beep/soap'>
S:     <![CDATA[<error code='550'>resource not
S:                                 supported</error>]]>
S: </profile>
```

Although the channel was created successfully, it remains in the "boot" state.

3. SOAP Message Packages

The BEEP profile for SOAP transmits envelopes encoded as UTF-8 using the media type "application/xml"[4], e.g.,

```
MSG 1 1 . 0 364
Content-Type: application/xml

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
       <m:GetLastTradePrice xmlns:m="Some-URI">
           <symbol>DIS</symbol>
       </m:GetLastTradePrice>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
END
```

In addition, the BEEP profile for SOAP also allows envelopes to be transmitted as the root part of a "multipart/related"[5] content, and with subordinate parts referenced using the rules of Section 3 of [6] (i.e., using either the "Content-ID:"[7] or "Content-Location:"[8] headers), e.g.,

```
MSG 1 2 . 364 668
Content-Type: multipart/related; boundary="MIME_boundary";
              type=application/xml;
              start="<claim061400a.xml@claiming-it.com>"

--MIME_boundary
Content-Type: application/xml
Content-ID: <claim061400a.xml@claiming-it.com>

<?xml version='1.0' ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
..
<theSignedForm href="cid:claim061400a.tiff@claiming-it.com" />
..
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--MIME_boundary
Content-Type: image/tiff
Content-Transfer-Encoding: binary
Content-ID: <claim061400a.tiff@claiming-it.com>

...binary TIFF image...
--MIME_boundary--
END
```

Consistent with Section 2 of [6], it is strongly recommended that the multipart contain a "start" parameter, and that the root part contain a "Content-ID:" header. However, because BEEP provides an 8bit-wide path, a "transformative" Content-Transfer-Encoding (e.g., "base64" or "quoted-printable") should not be used. Further note that MIME[9] requires that the value of the "Content-ID" header be globally unique.

4. SOAP Message Patterns

4.1 One-way Message

A one-way message involves sending a message without any response being returned.

The BEEP profile for SOAP achieves this using a one-to-many exchange, in which the client sends a "MSG" message containing an envelope, and the server immediately sends back a "NUL" message, before processing the contents of the envelope.

4.2 Request-Response Exchange

A request/response exchange involves sending a request, which results in a response being returned.

The BEEP profile for SOAP achieves this using a one-to-one exchange, in which the client sends a "MSG" message containing an envelope, and the server sends back a "RPY" message containing an envelope.

Finally, the BEEP profile for SOAP does not use the "ERR" message for SOAP faults when performing one-to-one exchanges -- whatever response is generated by the server is always returned in the "RPY" message.

4.3 Request/N-Responses Exchange

A request/N-responses exchange involves sending a request, which results in zero or more responses being returned.

The BEEP profile for SOAP achieves this using a one-to-many exchange, in which the client sends a "MSG" message containing an envelope, and the server sends back zero or more "ANS" messages, each containing an envelope, followed by a "NUL" message.

5. URL Schemes

This memo defines two URL schemes, "soap.beep" and "soap.beeps", which identify the use of SOAP over BEEP over TCP. Note that, at present, a "generic" URL scheme for SOAP is not defined.

5.1 The soap.beep URL Scheme

The "soap.beep" URL scheme uses the "generic URI" syntax defined in Section 3 of [10], specifically:

- the value "soap.beep" is used for the scheme component; and,
- the server-based naming authority defined in Section 3.2.2 of [10] is used for the authority component.
- the path component maps to the "resource" component of the boot message sent during profile initialization (if absent, it defaults to "/").

The values of both the scheme and authority components are case-insensitive.

For example, the URL

```
soap.beep://stockquoteserver.example.com/StockQuote
```

might result in the example shown in Section 2.1.

5.1.1 Resolving IP/TCP Address Information

The "soap.beep" URL scheme indicates the use of the BEEP profile for SOAP running over TCP/IP.

If the authority component contains a domain name and a port number, e.g.,

```
soap.beep://stockquoteserver.example.com:1026
```

then the DNS is queried for the A RRs corresponding to the domain name, and the port number is used directly.

If the authority component contains a domain name and no port number, e.g.,

```
soap.beep://stockquoteserver.example.com
```

the SRV algorithm[11] is used with a service parameter of "soap-beep" and a protocol parameter of "tcp" to determine the IP/TCP addressing information. If no appropriate SRV RRs are found (e.g., for "_soap-beep._tcp.stockquoteserver.example.com"), then the DNS is queried for the A RRs corresponding to the domain name and the port number used is assigned by the IANA for the registration in Section 7.4.

If the authority component contains an IP address, e.g.,

```
soap.beep://10.0.0.2:1026
```

then the DNS is not queried, and the IP address is used directly. If a port number is present, it is used directly; otherwise, the port number used is assigned by the IANA for the registration in Section 7.4.

While the use of literal IPv6 addresses in URLs is discouraged, if a literal IPv6 address is used in a "soap.beep" URL, it must conform to the syntax specified in [12].

5.2 The soap.beeps URL Scheme

The "soap.beeps" URL scheme is identical, in all ways, to the "soap.beep" URL scheme specified in Section 5.1, with the exception that prior to starting the BEEP profile for SOAP, the BEEP session must be tuned for privacy. In particular, note that both URL schemes use the identical algorithms and parameters for address resolution as specified in Section 5.1.1 (e.g., the same service name for SRV lookups, the same port number for TCP, and so on).

There are two ways to perform privacy tuning on a BEEP session, either:

- a transport security profile may be successfully started; or,
- a user authentication profile that supports transport security may be successfully started.

Regardless, upon completion of the negotiation process, a tuning reset occurs in which both BEEP peers issue a new greeting. Consult Section 3 of [2] for an example of how a BEEP peer may choose to issue different greetings based on whether privacy is in use.

6. Registration Templates

6.1 SOAP Profile Feature Registration Template

When a feature for the BEEP profile for SOAP is registered, the following information is supplied:

Feature Identification: specify a string that identifies this feature. Unless the feature is registered with the IANA, the feature's identification must start with "x-".

Feature Semantics: specify the semantics of the feature.

Contact Information: specify the electronic contact information for the author of the feature.

7. Initial Registrations

7.1 Registration: The SOAP Profile

Profile Identification: http://iana.org/beep/soap

Messages exchanged during Channel Creation: bootmsg, bootrpy

Messages starting one-to-one exchanges: bootmsg, SOAP-Env:Envelope

Messages in positive replies: bootrpy, SOAP-Env:Envelope

Messages in negative replies: error

Messages in one-to-many exchanges: SOAP-Env:Envelope

Message Syntax: SOAP-Env:Envelope as defined in Section 4 of [1] and [6]

Message Semantics: c.f., [1]

Contact Information: Eamon O'Tuathail <eamon.otuathail@clipcode.com>, Marshall Rose <mrose@dbc.mtview.ca.us>

7.2 Registration: The soap.beep URL Scheme

URL scheme name: soap.beep

URL scheme syntax: c.f., Section 5.1

Character encoding considerations: c.f., the "generic URI" syntax defined in Section 3 of [10]

Intended usage: identifies a SOAP resource made available using the BEEP profile for SOAP

Applications using this scheme: c.f., "Intended usage", above

Interoperability considerations: n/a

Security Considerations: c.f., Section 8

Relevant Publications: c.f., [1], [6], and [2]

Contact Information: Eamon O'Tuathail <eamon.otuathail@clipcode.com>, Marshall Rose <mrose@dbc.mtview.ca.us>

Author/Change controller: the IESG

7.3 Registration: The soap.beeps URL Scheme

URL scheme name: soap.beeps

URL scheme syntax: c.f., Section 5.2

Character encoding considerations: c.f., the "generic URI" syntax defined in Section 3 of [10]

Intended usage: identifies a SOAP resource made available using the BEEP profile for SOAP after the BEEP session has been tuned for privacy

Applications using this scheme: c.f., "Intended usage", above

Interoperability considerations: n/a

Security Considerations: c.f., Section 8

Relevant Publications: c.f., [1], [6], and [2]

Contact Information: Eamon O'Tuathail <eamon.otuathail@clipcode.com>, Marshall Rose <mrose@dbc.mtview.ca.us>

Author/Change controller: the IESG

7.4 Registration: The System (Well-Known) TCP port number for SOAP over BEEP

Protocol Number: TCP

Message Formats, Types, Opcodes, and Sequences: c.f., Section 2.1

Functions: c.f., [1]

Use of Broadcast/Multicast: none

Proposed Name: SOAP over BEEP

Short name: soap-beep

Contact Information: Eamon O'Tuathail <eamon.otuathail@clipcode.com>, Marshall Rose <mrose@dbc.mtview.ca.us>

8. Security Considerations

Although service provisioning is a policy matter, at a minimum, all implementations must provide the following tuning profiles:

for authentication: http://iana.org/beep/SASL/DIGEST-MD5

for confidentiality: http://iana.org/beep/TLS (using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher)

for both: http://iana.org/beep/TLS (using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher supporting client-side certificates)

Further, implementations may choose to offer MIME-based security services providing message integrity and confidentiality, such as OpenPGP[13] or S/MIME[14].

Regardless, consult [2]'s Section 9 for a discussion of BEEP-specific security issues.

References

[1] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H., Thatte, S. and D. Winer, "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>.

[2] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, March 2001.

[3] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[4] Murata, M., St.Laurent, S. and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[5] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, August 1998.

[6] Barton, J., Thatte, S. and H. Nielsen, "SOAP Messages with Attachments", December 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211>.

[7] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.

[8] Palme, F., Hopmann, A., Shelness, N. and E. Stefferud, "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)", RFC 2557, March 1999.

[9] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[10] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[11] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[12] Haskin, D. and E. Allen, "IP Version 6 over PPP", RFC 2472, December 1998.

[13] Elkins, M., Del Torto, D., Levien, R. and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

[14] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

IANA Considerations

The IANA has registered the profile specified in Section 7.1 as:

http://iana.org/beep/soap

The IANA has registered "soap.beep" and "soap.beeps" as URL schemes, as specified in Section 7.2 and Section 7.3, respectively.

The IANA has also registered "SOAP over BEEP" as a TCP port number, as specified in Section 7.4.

Finally, the IANA maintains a list of SOAP profile features, c.f., Section 6.1. The IESG is responsible for assigning a designated expert to review the specification prior to the IANA making the assignment. Prior to contacting the IESG, developers of SOAP profile features must use the mailing list beepwg@lists.beepcore.org to solicit commentary.

Acknowledgements

The authors gratefully acknowledge the contributions of: Christopher Ferris, Huston Franklin, Alexey Melnikov, Bill Mills, and Roy T. Fielding.

Authors' Addresses

Eamon O'Tuathail
Clipcode.com
24 Thomastown Road
Dun Laoghaire
Dublin
IE

Phone: +353 1 2350 424
EMail: eamon.otuathail@clipcode.com
URI: http://www.clipcode.com/

Marshall T. Rose
Dover Beach Consulting, Inc.
POB 255268
Sacramento, CA 95865-5268
US

Phone: +1 916 483 8878
EMail: mrose@dbc.mtview.ca.us

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.