FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Agent Message Transport Protocol for HTTP Specification

Document title	FIPA Agent Message Transport Protocol for HTTP Specification		
Document number	XC00084E~~D~~	**Document source**	FIPA Agent Management
Document status	Experimental	**Date of this status**	2001/10/18~~08/10~~
Supersedes	None		
Contact	fab@fipa.org		
Change history			
2000/08/24	Approved for Experimental		
2001/08/10	Line numbering added		
2002/07/25	See *Informative Annex C — ChangeLog*		

Geneva, Switzerland

Notice

Use of the technologies described in this specification may infringe patents, copyrights or other intellectual property rights of FIPA Members and non-members. Nothing in this specification should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. FIPA strongly encourages anyone implementing any part of this specification to determine first whether part(s) sought to be implemented are covered by the intellectual property of others, and, if so, to obtain appropriate licenses or other permission from the holder(s) of such intellectual property prior to implementation. This specification is subject to change without notice. Neither FIPA nor any of its Members accept any responsibility whatsoever for damages or liability, direct or consequential, which may result from the use of this specification.

consequential, which may result from the use of this specification.

Foreword

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. This occurs through open collaboration among its member organizations, which are companies and universities that are active in the field of agents. FIPA makes the results of its activities available to all interested parties and intends to contribute its results to the appropriate formal standards bodies where appropriate.

The members of FIPA are individually and collectively committed to open competition in the development of agent-based applications, services and equipment. Membership in FIPA is open to any corporation and individual firm, partnership, governmental body or international organization without restriction. In particular, members are not bound to implement or use specific agent-based standards, recommendations and FIPA specifications by virtue of their participation in FIPA.

The FIPA specifications are developed through direct involvement of the FIPA membership. The status of a specification can be either Preliminary, Experimental, Standard, Deprecated or Obsolete. More detail about the process of specification may be found in the FIPA Document Policy [f-out-00000] and the FIPA Specifications Policy [f-out-00003]~~Procedures for Technical Work~~. A complete overview of the FIPA specifications and their current status may be found ~~in the FIPA List of Specifications. A list of terms and abbreviations used in the FIPA specifications may be found in the FIPA Glossary~~on the FIPA Web site.

FIPA is a non-profit association registered in Geneva, Switzerland. As of June~~anuary~~ 2002~~0~~, the 56 members of FIPA represented many~~17~~ countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found on the FIPA Web site at http://www.fipa.org/.

Contents

1 Scope

This document is part of the FIPA specifications and deals with message transportation between inter-operating agents. This document also forms part of the FIPA Agent Management Specification [FIPA00023] and contains specifications for:

- The transportation of messages between agents using the Hypertext Transfer Protocol (HTTP - see [RFC2616]).

2 Message Transport Protocol for HTTP

This MTP is based on the transfer of data representing the entire agent message including the message envelope in a HTTP request. The HTTP data transfer is a two-step process: the sender makes a HTTP request and after receiving the data the receiver sends a HTTP response. The receiver then parses the message envelope and the message is handled according to the instructions and information given in the message envelope.

2.1 Component Name

The name assigned to this component is:

```
fipa.mts.mtp.http.std
```

2.2 Interface Definition

2.2.1 Request

A HTTP request comprises:

- **Request Line**
  - The request method type that must be `POST`.
  - The request resource identification that must be a full URI (see [RFC1630]).
  - The request version that must be `HTTP/1.1`.
- **Request Headers**
  - The mandatory parameter `Content-Type:` that must be "multipart/mixed" and must have a `boundary` parameter enclosed by double quotes. It should be anticipated that the boundary parameter may be "folded" as described in [RFC2822] – hence parsers must be able to handle this type of encoding.
  - The mandatory parameter `Host:` that must be in the form *`hostname`* or *`hostname`*`:`*`portnumber`*.
  - The mandatory parameter `Cache-Control:` that must have the value `no-cache`.
  - The mandatory parameter `MIME-Version:` that must have the value 1.0.
  - The optional parameter `Content-Length:` that contains the size of the request body[1].
- **Request Body**

  The request body contains the agent message. The agent message has two components (separated as defined in [RFC2046] for multipart/mixed MIME content): a FIPA message envelope and a FIPA message body (the payload).

  The encoded body must therefore contain at least two parts, the first part containing the FIPA message envelope, the second part containing the FIPA Message being sent. Each of the two parts must specify an encoding-level `Content-Type` field which may be any MIME type (Implementations must assume that some parts of the multipart encoded content may contain raw binary data). Each of the two parts may contain other headers such as, for example, `Content-Transfer-Encoding` but the processing of these fields is not mandatory.

  The `charset` used in headers and the boundary delimiter of the multipart encoding must be plain ASCII.

[1] See [RFC2616] which strongly recommends that this parameter is used.

Where applicable the `charset` encoding of the FIPA Message must be specified as a `charset` parameter of the `Content-Type` header. This `charset` parameter value must have the same value as the value of the envelope `payload-encoding` field.

~~The parts encoded in the multipart message body are enclosed between boundary delimiters. The boundary delimiter is formed from the boundary value specified as parameter for the `ContentType` header. The boundary value must be a sequence of maximum 70 ASCII chars. Each MIME part is to be considered enclosed between two occurrences of the sequence "CRLF--boundary value". The last boundary delimiter must be a boundary delimiter ending line and is formed from the usual boundary delimiter followed by the sequence "--", that is, "CRLF--boundary value--".~~

The envelope body encoding must therefore have the following structure:

- MIME headers (at least a MIME-Version header and a `Content-Type` header that contains the boundary value).

- An empty line delimiting the MIME headers from the MIME body.

- A boundary delimiter line that delimits the beginning of the envelope part.

- A `Content-Type` header line that must have the value appropriate for the envelope representation (`“application” “/”` <string>, where the string is the component name given in each envelope specification).

- An empty line (CRLF CRLF).

- The FIPA message envelope.

- A boundary delimiter line that delimits the FIPA envelope from the FIPA message.

- A `Content-Type` header line that must have the value appropriate for the FIPA Message representation (`“application” “/”` <string>, where the string is the component name given in each message encoding specification).~~.~~

- A boundary delimiter line that defines the end of the FIPA Message. This boundary line MAY be a boundary delimiter ending line.

2.2.2 Response

A HTTP response comprises:

- **Response Line**
  The response version must be `HTTP/1.1`. The response status code must either be the success code or a suitable error code as defined in [RFC2616]. The success code only means that the receiving agent has succeeded in extracting the message content from the HTTP request. More detailed information about non-HTTP related issues such as envelope parsing and message handling should be sent back to the sender agent as a separate message. If a sending MTP receives an error code then the expected behaviour would be to try sending the message using another combination of target resource address and content type or give up. The reason phrase in any error response may be any string and is used only for informational purposes.

- **Response Headers**
  - The mandatory parameter `Content-Type:` can be any MIME type (see [RFC2045])

  - The mandatory parameter `Cache-Control:` must have the value `no-cache`, and

  - The optional parameter `Content-Length:` specifies the size of the response body[2]

- **Response Body**
  The response body may contain a message reply and depending on the content type can be text, binary or multipart. The sender is not obliged to read or make use of such content (i.e. it should not be relied upon for message transfer).

2.2.3 Notes

The default connection behaviour on HTTP version 1.1 is to have persistent connections which means that after a request-response cycle, the connection is kept open and other requests can be made. However, because this would require a more complex implementation, connection persistence is not mandatory. In the case of a simple MTP implementation that would not support persistence, the `Connection:` parameter with the value `close` must be sent in the request headers if the MTP is acting as a sender or in the response headers if the MTP is acting as a receiver.

It should be anticipated that some of the header field values (especially the boundary parameter of the Content-Type request field) are "folded" as described in [RFC2822]. So parsers must be able to handle this type of encoding.

Compliance to the MTP described in this document does not require HTTP 1.1 features that are not explicitly mentioned here.

~~2.3Envelope Syntax~~

~~The syntax used for the representation of the FIPA message envelope is that defined in [FIPA00085].~~

[2] See [RFC2616] which strongly recommends that this parameter is used.

~~2.4 Notes for Developers~~

~~1.The boundary field is usually “folded” on a new line. So the underlying system should be able to fold/unfold encoded MIME headers and values.~~

~~2.In the MIME body before each boundary delimiter there must be a new line separator that is considered to be part of the boundary delimiter. So sections are delimited by the sequence "CRLF--boundary value" (where CRLF are two octets with values of 13 and 10 representing the ASCII characters CR and LF, boundary value is the sequence specified in the `ContentType` value as parameter, and “--“ are two ASCII minus characters).~~

~~3.Good implementations will generate random boundary values and will check that none of the encoded parts contains the boundary delimiter sequence.~~

~~4.It is possible to have some text before the first boundary delimiter line and after the ending boundary delimiter line, namely a prologue and an epilogue. This text is to be ignored and should be there only to emphasise the boundary delimiters.~~

~~2.5~~2.3 References

[FIPA00023] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. `http://www.fipa.org/specs/fipa00023/`

[FIPA00067] FIPA Agent Message Transport Service Specification. Foundation for Intelligent Physical Agents, 2000. `http://www.fipa.org/specs/fipa00067/`

~~[FIPA00085] FIPA Agent Message Transport Envelope Representation in XML. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00085/~~

[RFC2822] Standard for the Format of ARPA Internet Text Messages. Request for Comments, 2001~~1982~~. http://www.ietf.org/rfc/rfc2~~0~~822.txt

[RFC1630] Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World Wide Web. Request for Comments, 1994. `http://www.ietf.org/rfc/rfc1630.txt`

[RFC2045] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Request for Comments, 1996. `http://www.ietf.org/rfc/rfc2045.txt`

[RFC2046] Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. Request for Comments, 1996. `http://www.ietf.org/rfc/rfc2045.txt`

[RFC2616] Hypertext Transfer Protocol - HTTP/1.1. Request for Comments, 1999. `http://www.ietf.org/rfc/rfc2616.txt`

3 Informative Annex A — Example

The agent `sender@bar.com` sends a message to the agent `receiver@foo.com` which is resident on an AP that has an ACC with an external HTTP interface. Both agents are simple implementations that do not use connection persistence and the message encoding (see [FIPA00085]) that they use is text.

1. `sender@bar.com` sends a message to `receiver@foo.com`:

```
POST http://foo.com:80/acc HTTP/1.1
Cache-Control: no-cache
Host: foo.com:80
Mime-Version: 1.0
Content-Type: multipart-mixed ;
      boundary="251D738450A171593A1583EB"
Content-Length: 1518
Connection: close[3]

This is not part of the MIME multipart encoded message.
--251D738450A171593A1583EB
Content-Type: application/fipa.mts.env.rep.xml.std~~xml~~

<?xml version="1.0"?>
<envelope>
  <params index="1">
    <to>
      <agent-identifier>
        <name>receiver@foo.com</name>
        <addresses>
          <url>http://foo.com/acc</url>
        </addresses>
      </agent-identifier>
    </to>
    <from>
      <agent-identifier>
        <name>sender@bar.com</name>
        <addresses>
          <url>http://bar.com/acc</url>
        </addresses>
      </agent-identifier>
    </from>

    <acl-representation>fipa.acl.rep.string.std</acl-representation>

    <payload-encoding>US-ASCII</payload-encoding>

    <date>20000508T042651481</date>

~~    <encrypted>no encryption</encrypted>~~

    <received >
      <received-by value="http://foo.com/acc" />
      <received-date value="20000508T042651481" />
      <received-id value="123456789" />
    </received>
  </params>
</envelope>[4]

--251D738450A171593A1583EB
Content-Type: application/fipa.acl.rep.string.std~~application/text~~; charset=US-ASCII

```

[3] Followed by an empty line.

[4] CRLF at the end of the XML Envelope

```
(inform
  :sender
    (agent-identifier
      :name sender@bar.com
      :addresses (sequence http://bar.com:80/acc))
  :receiver
    (agent-identifier
      :name receiver@foo.com
      :addresses (sequence http://foo.com:80/acc )) )
  :content-length 14
  :reply-with task1-003
  :language FIPA-sl0
  :ontology planning-ontology-1
  :content
    “((done task1)))”

--251D738450A171593A1583EB--
```

2. The ACC responds with a successful notification:

```
HTTP/1.1 200 OK
Content-Type: text/plain
Cache-Control: no-cache
Connection: close[5]
```

[5] Followed by an empty line.

4 Informative Annex A — Notes for Developers

1. The boundary field is usually “folded” on a new line. So the underlying system should be able to fold/unfold encoded MIME headers and values.

2. In the MIME body before each boundary delimiter there must be a new line separator that is considered to be part of the boundary delimiter. So sections are delimited by the sequence "CRLF--boundary value" (where CRLF are two octets with values of 13 and 10 representing the ASCII characters CR and LF, boundary value is the sequence specified in the `Content-Type` value as parameter, and “--“ are two ASCII minus characters).

3. Good implementations will generate random boundary values and will check that none of the encoded parts contains the boundary delimiter sequence.

4. It is possible to have some text before the first boundary delimiter line and after the ending boundary delimiter line, namely a prologue and an epilogue. This text is to be ignored and should be there only to emphasise the boundary delimiters.

5. [RFC2616]: “In the interest of robustness, servers SHOULD ignore any empty line(s) received where a Request-Line is expected. In other words, if the server is reading the protocol stream at the beginning of a message and receives a CRLF first, it should ignore the CRLF.

   Certain buggy HTTP/1.0 client implementations generate extra CRLF's after a POST request. To restate what is explicitly forbidden by the BNF, an HTTP/1.1 client MUST NOT preface or follow a request with an extra CRLF.”

6. In order to facilitate the dynamic discovery of remote platforms, it is recommended, but not strictly mandated, to launch the HTTP-MTP server at the following URL http://<host name>:80/fipa.mts i.e. using 80 as port number and fipa.mts as target.

5 Informative Annex C — ChangeLog

5.1 2002/07/25 — version E by X2S TC

Entire specification:	Changed “ContentType” header field to “Content-Type”.
Page 3, Lines 110-115:	Removed paragraph related to MIME boundaries.
Page 3, Line 126:	**Changed the envelope part Content-Type to enable use of any FIPA specified envelope encoding.**
Page 3, Line 135:	Clarification to message part Content-Type definition.
Page 4, Line 177:	Removed unnecessary and incorrect Section about envelope encoding.
Page 4, Lines 180-194:	Moved the section as an informative appendix.
Page 5, Line 200:	Removed reference to specification number 85.
Page 6, Line 262:	Removed “encrypted” envelope header field.
Page 6, Line 234:	Corrected the Content-Type header field value.
Page 7, Line 273:	Corrected the Content-Type header field value.
Page 7, Line 289:	Added quotes to ACL content.
Page 9, line 330-332:	**Added note on recommended URL for HTTP-MTP**