FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Agent Message Transport Envelope Representation in String Specification

Document title	FIPA Agent Message Transport Envelope Representation in String Specification		
Document number	DC00073D	**Document source**	FIPA Agent Management
Document status	Deprecated	**Date of this status**	2001/08/10
Supersedes	FIPA00024		
Contact	fab@fipa.org		
Change history			
2000/07/20	Deprecated since current MTPs no longer use string based envelopes		
2001/08/10	Line numbering added		

Geneva, Switzerland

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.

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 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.

FIPA is a non-profit association registered in Geneva, Switzerland. As of January 2000, the 56 members of FIPA represented 17 countries worldwide. Further information about FIPA as an organization, membership information, FIPA specifications and upcoming meetings may be found 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:

Syntactic representation of a message envelope in string form.

2 String Envelope Representation

This section gives the concrete syntax for the message envelope in string format. This concrete syntax and lexical analysis of messages has been inspired by [RFC822].

2.1 Component Name

The name assigned to this component is:

```
fipa.mts.env.rep.string.std
```

2.2 Lexical Analysis

Messages consist of message envelope parameters and, optionally, a message body. The message body is simply a sequence of ASCII characters representing an ACL message. The message body is separated from the message envelope by two subsequent CRLF tokens with nothing in between the tokens (that is, a line with nothing preceding the CRLF).

Each message envelope parameter can be viewed as a single, logical line of ASCII characters, comprising a parameter name and a parameter value. For convenience, the parameter value portion of this conceptual entity can be split into a multiple-line representation by inserting, at the transmitter side, a CRLF immediately followed by at least one LWSP-char (this action is called *folding*). At the receiver side, CRLF immediately followed by a LWSP-char is considered equivalent to the LWSP-char (this action is called *unfolding*).

Once a parameter has been unfolded, at the receiver side it may be viewed as being composed of a parameter name, followed by a colon (`:`), followed by a parameter body, and terminated by a carriage-return/line-feed (CRLF). The parameter name must be composed of printable ASCII characters (that is, characters that have values between 33 and 126 decimal, except colon). The parameter body may be composed of any ASCII characters, except CR or LF. (While CR and/or LF may be present in the actual text, they are removed by the action of unfolding the parameter.)

Except as noted, alphabetic strings may be represented in any combination of upper and lower case. However, ACCs are required to preserve case information when transporting messages.

These rules show a parameter meta-syntax, without regard for the particular type or internal syntax. Their purpose is to permit detection of parameters; also, they present to higher-level parsers an image of each parameter as fitting on one line.

```
MessageEnvelope         = Parameter+ CRLF MessageBody.

MessageBody             = Text* ( CRLF Text* )*
                        | Byte*.[1]

Parameter               = ParameterName ":" [ ParameterBody ] CRLF.

ParameterName           = 1* <any CHAR, excluding CTLs, SPACE, and ":">.

ParameterBody           = ParameterBodyContents [CRLF LWSP-char ParameterBody].

ParameterBodyContents   = <the ASCII characters making up the ParameterBody, as defined
                           in the following section and consisting of combinations of
                           Atom, QuotedString and specials tokens or else consisting of
                           Text>.
```

[1] Note that this cannot be transmitted over [FIPA00075].

The following rules are used to define an underlying lexical analyser, which feeds tokens to higher-level parsers.

```
                                                    ; (  Octal, Decimal.)

CHAR                    = <any ASCII character>.    ; (  0-177,  0.-127.)

DIGIT                   = <any ASCII decimal digit>.; ( 60- 71, 48.- 57.)

CTL                     = <any ASCII control        ; (  0- 37,  0.- 31.)
                           character and DEL>.      ; (    177,     127.)

CR                      = <ASCII CR, carriage return>.  ; (     15,      13.)

LF                      = <ASCII LF, linefeed>.     ; (     12,      10.)

SPACE                   = <ASCII SP, space>.        ; (     40,      32.)

HTAB                    = <ASCII HT, horizontal-tab>.   ; (     11,       9.)

<">                     = <ASCII quote mark>.       ; (     42,      34.)

CRLF                    = CR LF.

LWSPChar                = SPACE / HTAB.             ; semantics = SPACE

LinearWhiteSpace        = ([CRLF] LWSPChar)+.       ; semantics = SPACE
                                                    ; CRLF => folding

Text                    = <any CHAR including bare CR and
                           bare LF but NOT including CRLF>.

Atom                    = <any CHAR except <">, SPACE and CTLs>
                          <any CHAR except SPACE and CTLs> *.

QuotedString            = <"> ( QText/QuotedPair )* <">.    ; Regular qtext or
                                                    ; quoted chars.

QText                   = <any CHAR excepting <">,          ; => may be folded
                               "\" and CR, and including linear-white-space>.

QuotedPair              = "\" CHAR.                 ; may quote any char

Word                    = Atom / QuotedString.

Byte                    = <any 8-bit byte>.
```

2.3 Syntax

The following rules apply after the unfolding operation, as specified in the previous section.

```
MessageEnvelope             = Parameter+ CRLF MessageBody.

Parameter                   = ACLRepresentationParameter  CRLF
                            | CommentParameter            CRLF
                            | ContentLengthParameter      CRLF
                            | ContentEncodingParameter    CRLF
                            | DateParameter               CRLF
                            | EncryptedParameter          CRLF
                            | IntendedReceiverParameter   CRLF
                            | ReceivedParameter           CRLF
                            | EnvSenderParameter          CRLF
                            | EnvReceiverParameter        CRLF
```

```
                              | TransportBehaviourParameter CRLF
                              | UserDefinedParameter       CRLF.


MessageBody                   = Text* ( CRLF Text* )*
                              | CRLF Byte*.[2]

ACLRepresentationParameter    = "ACL-representation" ":" word.

CommentParameter              = "Comments" ":" text*.

ContentLengthParameter        = "Content-length" ":" DIGIT+.

ContentEncodingParameter      = "Content-encoding" ":" word.

DateParameter                 = "Date" ":" DateTime.

DateTime                      = See section 2.5.

EncryptedParameter            = "Encrypted" ":" word [ word ].

IntendedReceiverParameter     = "Intended-receiver" ":" AgentIdentifierList.

AgentIdentifierList           = AgentIdentifier [ "," AgentIdentifier ]*.

ReceivedParameter             = "Received" ":"
                                [ "from" URL ]
                                [ "by"   URL ]
                                [ "id"   word ]
                                [ "via"   word ]
                                ";" DateTime.

EnvSenderParameter            = "From" ":" AgentIdentifier.

EnvReceiverParameter          = "To" ":" AgentIdentifierList.

TransportBehaviourParameter   = "Transport-behaviour" ":"
                                [ "error-messages" AgentIdentifierList ]
                                [ "delivery" word ]
                                [ "acknowledgement" AgentIdentifierList ].

UserDefinedParameter          = <any parameter which has not been defined in this
                                 specification or published as an extension to this
                                 specifications; parameter name must be unique and may
                                 be pre-empted by published extensions.>.

AgentIdentifier               = "(" "AID"
                                    ":name" Word
                                  [ ":addresses" URLSequence ]
                                  [ ":resolvers" AgentIdentifierSequence ]
                                  ( UserDefinedParameter Expression )* ")".

AgentIdentifierSequence       = "(" "sequence" AgentIdentifier* ")".[3]

URLSequence                   = "(" "sequence" URL* ")".

URL                           = See [RFC2396]
```

[2] Note that this cannot be transmitted over [FIPA00075].

[3] A sequence is considered to have a left to right (first to last) ordering.

2.4 Additional Syntax Rules

The following additional rules not specified in the grammar also apply:

1. The abstract syntax of the message envelope is mandatory.

2. This specification permits multiple occurrences of message envelope parameters. For the purposes of disambiguation the first occurrence overrides any subsequent occurrence (see [RFC822] for further details).

In the future, additional parameters may be defined and added to the message envelope. Such parameters are prefixed with `X-FIPA-` and their behaviour is not specified. If an organisation wishes to add its own message envelope parameters it is suggested they prefix the new parameter name with `X-CompanyName-` to reduce the chances of conflict.

2.5 Representation of Time

Time tokens are based on [ISO8601], with extensions for relative time and millisecond durations. Time expressions may be absolute, or relative to the current time. Relative times are distinguished by the character + appearing as the first character in the construct. If no type designator is given, the local time zone is used. The type designator for UTC is the character `Z`. UTC is preferred to prevent time zone ambiguities. Note that years must be encoded in four digits. As examples, 8:30am on April 15th, 1996 local time would be encoded as:

```
19960415T083000000
```

The same time in UTC would be:

```
19960415T083000000Z
```

While one hour, 15 minutes and 35 milliseconds from now would be:

```
+00000000T011500035
```

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/

[FIPA00075] FIPA Agent Message Transport Protocol for IIOP Specification. Foundation for Intelligent Physical Agents, 2000. http://www.fipa.org/specs/fipa00075/

[ISO8601] Date Elements and Interchange Formats, Information Interchange-Representation of Dates and Times. International Standards Organisation, 1998. http://www.iso.ch/cate/d15903.html

[RFC822] Uniform Resource Identifiers: Generic Syntax. Request for Comments, 1992. http://www.ietf.org/rfc/rfc0822.txt

[RFC2396] Standard for the Format of APRA Internet Text Messages. Request for Comments, 1998. http://www.ietf.org/rfc/rfc2396.txt