FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Agent Message Transport Envelope Representation in Bit-Efficient Encoding Specification

Document title	FIPA AMT Envelope	PA AMT Envelope Representation in Bit-Efficient Encoding Specification									
Document number	XC00088B	Document source	FIPA Agent Management								
Document status	Experimental	Date of this status	2001/08/10								
Supersedes	None										
Contact	fab@fipa.org										
Change history											
2001/08/10	Approved for Experin	mental; Line numbering added									

- 18 © 2000 Foundation for Intelligent Physical Agents http://www.fipa.org/
- 19 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.

20 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 agentbased 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 agentbased 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/.

39	Contents

40	1 Sc	ope	. 1
41	2 Bit-	Efficient Envelope Representation	. 2
42	2.1	Component Name	. 2
43	2.2	ACC Processing of Bit-Efficient Envelope	. 2
44	2.3	Concrete Message Envelope Syntax	. 3
45	2.4	Notes on the Grammar Rules	. 5
46	3 Exa	amples	. 7
47	4 Re	ferences	11
48			

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 bit-efficient form.
- 55 Informative examples of the bit-efficient envelope syntax are given in Section 3, Examples.

57 2 Bit-Efficient Envelope Representation

58 This section gives the concrete syntax for the message envelope specification that must be used to transport messages 59 over a Message Transport Protocol (MTP - see [FIPA00067]). This concrete syntax is designed to complement 60 [FIPA00069].

- 61
- The message envelope transport syntax is expressed in standard EBNF format (see Table 1).
- 62 63

Grammar rule component	Example
Terminal tokens are enclosed in double quotes	"("
Non-terminals are written as capitalised identifiers	Expression
Square brackets denote an optional construct	["," OptionalArg]
Vertical bars denote an alternative between choices	Integer Float
Asterisk denotes zero or more repetitions of the preceding expression	Digit*
Plus denotes one or more repetitions of the preceding expression	Alpha+
Parentheses are used to group expansions	(A B)*
Productions are written with the non-terminal name on the left-hand side,	ANonTerminal = "terminal".
expansion on the right-hand side and terminated by a full stop	
0x?? is a hexadecimal byte	0x00

64

65 66 Table 1: EBNF Rules

67 N.B. White space is not allowed between tokens.

68

79

69 2.1 Component Name

70 The name assigned to this component is:

```
71
72 fipa.mts.env.rep.bitefficient.std
73
```

74 2.2 ACC Processing of Bit-Efficient Envelope

According to [FIPA00067], a FIPA compliant ACC is not allowed to modify any element of the envelope that it receives. It is however allowed to update a value in any of the envelope's slots by adding a new ExtEnvelope element at the beginning of the messageEnvelopes sequence. This new element is required to have only those slot values that the ACC wishes to add or update plus a new ReceivedObject element¹.

80 The following pseudo code algorithm may be used to obtain the latest values for each of the envelope's slots.

```
81
82
     EnvelopeWithAllSlots := new empty Envelope
     while (not all envelopes processed) {
83
84
           tempEnvelope = getNextEnvelope;
85
           foreach slot in an envelope {
86
                  if ((this slot has no value in EnvelopeWithAllSlots)
87
                         AND (this slot has a value in tempEnvelope))
88
                  then copy the value of this slot to EnvelopeWithAllSlots
89
            }
90
     }
91
92
     EnvelopeWithAllSlots now contains the latest values for all the slots set in the envelope.
93
```

¹ The new ReceivedObject is forced, syntactically, to be in all envelopes of the messageEnvelopes sequence except the first one.

93 2.3 Concrete Message Envelope Syntax

94 95	MessageEnvelope	= (ExtEnvelope)* BaseEnvelope Pavload.
96 97		- RegeFrueleneWeeder (Slet)* EndofFruelene
97 98	BaseEnverope	= BaseEnveropeHeader (Siot)" EndorEnverope.
99 100	ExtEnvelope	= ExtEnvelopeHeader (Slot)* EndOfEnvelope.
101 102	BaseEnvelopeHeader	= BaseMsgId EnvLen ACLRepresentation Date.
103 104	ExtEnvelopeHeader	= ExtMsgId EnvLen ReceivedObject.
105 106 107	EnvLen	= Len16 JumboEnvelope. /* See comment 1 (Section 2.4) */
108 109	JumboEnvelope	= EmptyLen16 Len32.
110 111	BaseMsgId	= 0xfe.
112 113	ExtMsgId	$= 0 \times FD.$
114 115	EndOfEnvelope	= EndOfCollection.
116 117	Payload	= /* See comment 2 (Section 2.4) */
118 119 120	Slot	= PredefinedSlot UserDefinedSlot. /* See comment 5 (Section 2.4) */
121 122 123 124 125 126 127 128 129 130 131	PredefinedSlot	<pre>= 0x02 AgentIdentifierSequence /* to */ 0x03 AgentIdentifier /* from */ 0x04 ACLRepresentation /* acl-representation */ 0x05 Comments /* comments */ 0x06 PayloadLength /* payload-length */ 0x07 PayloadEncoding /* payload-encoding */ 0x08 Encrypted /* encrypted */ 0x09 IntendedReceiver /* intended-receiver */ 0x0a ReceivedObject /* received */ 0x0b TransportBehaviour. /* transport-behaviour */</pre>
132 133 134 135 136	ACLRepresentation	<pre>= UserDefinedACLRepresentation 0x10</pre>
137 138	Date	= BinDateTimeToken.
139 140	Comments	= NullTerminatedString.
141 142	PayloadLength	= BinNumber.
143 144	PayloadEncoding	= NullTerminatedString.
145 146	Encrypted	= StringSequence.
147 148	IntendedReceiver	= AgentIdentifierSequence.
149 150	TransportBehaviour	= Any.
151 152 153	UserDefinedACLRepreser	tation = 0x00 NullTerminatedString.
154	ReceivedObject	= By

155 156 157 158 159			Date [From] [Id] [Via] EndOfCollection.
160 161 162	Ву	=	URL.
162 163	From	=	0x02 URL.
165 166	Id	=	0x03 NullTerminatedString.
167 168	Via	=	0x04 NullTerminatedString.
169 170	BinNumber	=	Digits. /* See comment 4 (Section 2.4) */
171	Digits	=	CodedNumber+.
173	NullTerminatedString	=	String 0x00.
175	UserDefinedSlot	=	0x00 Keyword NullTerminatedString.
177	KeyWord	=	NullTerminatedString.
179 180 181	Any	= 	0x14 NullTerminatedString ByteLenEncoded.
182 183 184 185	ByteLenEncoded	=	0x16 Len8 ByteSequence 0x17 Len16 ByteSequence 0x19 Len32 ByteSequence.
186 187	ByteSequence	=	Byte*.
188	AgentIdentifierSequence	=	(AgentIdentifier)* EndOfCollection.
199 190 191 192 193 194 195	AgentIdentifier	=	0x02 AgentName [Addresses] [Resolvers] (UserDefinedParameter)* EndOfCollection.
196 197	AgentName	=	NullTerminatedString.
198 199	Addresses	=	0x02 UrlSequence.
200	Resolvers	=	0x03 AgentIdentifierSequence.
202	UserDefinedParameter	=	0x04 NullTerminatedString Any.
203	UrlSequence	=	(URL)* EndOfCollection.
205	URL	=	NullTerminatedString.
208	StringSequence	=	(NullTerminatedString)* EndOfCollection.
203 210 211 212	BinDateTimeToken	= 	0x20 BinDate 0x21 BinDate TypeDesignator.
212 213	BinDate	=	Year Month Day Hour Minute Second Millisecond.
∠14 215 216	EndOfCollection	=	0x01.
217 218	EmptyLen16	=	0x00 0x00.

219	Len8	= Byte.	/* S	ee	comment	6	(Section	2.4)	*/						
220	Len16	= Short.	/* S	ee	comment	6	(Section	2.4)	*/						
223	Len32	= Long.	/* S	ee	comment	6	(Section	2.4)	*/						
224 225 226	Year	= Byte Byte.													
220 227 228	Month	= Byte.													
229 230	Day	= Byte.													
231 232	Hour	= Byte.													
233 234	Minute	= Byte.													
235 236	Second	= Byte.													
237 238	Millisecond	= Byte Byte.													
239 240	String	= /* As in [FIPA00	070]	*/											
241 242	CodedNumber	= /* See comment 4	1 (Se	cti	on 2.4)	*/	,								
243 244	TypeDesignator	= /* As in [FIPA00	070]	*/											

245 **2.4 Notes on the Grammar Rules**

253

257

261

265

- Normally, the length of an envelope does not exceed 65536 bytes (2^16). Therefore, only two bytes are reserved for envelope length (len16). However, the syntax also allows envelopes with greater lengths. In this case, the sender sets the reserved envelope length slot (two bytes) to length zero, and the following four bytes are used to represent the real length (maximum envelope length is therefore 2^32 bytes).
- The length of the envelope comprises all the parts of the envelope, including the message identifier and the length slot itself. The length of the envelope is expressed in the network byte order.
- The payload (ACL message) starts at the first byte after the BaseEnvelope. White space is allowed between the
 envelope and the ACL message only if the syntax of ACL allows this. For instance, fipa.acl.rep.string.std
 allows white space, but fipa.acl.rep.bitefficient.std does not.
- Dates are coded as numbers, that is, four bits are reserved for each ASCII number (see comment 4 below).
 Information as to whether the type designator is present or not is coded into an identifier byte. These slots always
 have static length (two bytes for year and milliseconds, one byte for other components).
- 4. Numbers are coded by reserving four bits for each digit in the number's ASCII representation, that is, two ASCII numbers are coded into one byte. *Table 2* shows a 4-bit code for each number and special codes that may appear in ASCII coded numbers.
- 266 If the ASCII presentation of a number contains an odd number of characters, the last four bits of the coded number 267 are set to zero (the Padding token), otherwise an additional 0×00 byte is added to the end of the coded number. If 268 the number to be coded is either an integer, decimal number, or octal number, the identifier byte 0×12 is used. For 269 hexadecimal numbers, the identifier byte 0×13 is used. Hexadecimal numbers are converted to integers before 270 coding (the coding scheme does not allow characters from a through f to appear in number form). 271

Token	Code	Token	Code
Padding	0000	7	1000
0	0001	8	1001

1	0010	9	1010
2	0011	+	1100
3	0100	Е	1101
4	0101	-	1110
5	0110	•	1111
6	0111		

272

273 274 Table 2: Binary Representation of Number Tokens

- 5. All envelope parameters defined in [FIPA00067] have a predefined code. If an envelope contains a user-defined parameter, an extension mechanism is used (byte 0x00). The names of the user-defined envelope parameters should have the prefix "X-CompanyName-".
- Byte is a one-byte code word, Short is a short integer (two bytes, network byte order) and Long is a long integer
 (four bytes, network byte order).
- 281

Examples 3 281

282

284

```
283
       1. Here is a simple example of an envelope encoded using XML representation:
```

```
285
      <?xml version="1.0"?>
286
       <envelope>
287
         <params index="1">
288
           <to>
289
              <agent-identifier>
290
                <name>receiver@foo.com</name>
291
                <addresses>
292
                  <url>http://foo.com/acc</url>
293
                </addresses>
294
              </agent-identifier>
295
           </to>
296
           <from>
297
              <agent-identifier>
298
                <name>sender@bar.com</name>
299
                <addresses>
300
                  <url>http://bar.com/acc</url>
301
                </addresses>
302
              </agent-identifier>
303
           </from>
304
305
           <acl-representation>fipa.acl.rep.xml.std</acl-representation>
306
307
           <date>20000508T042651481</date>
308
309
           <encrypted>no encryption</encrypted>
310
311
           <received>
312
              <received-by value="http://foo.com/acc" />
313
              <received-date value="20000508T042651481" />
314
              <received-id value="123456789" />
315
           </received>
316
         </params>
317
      </envelope>
318
319
      Using the bit-efficient representation, the envelope becomes:
320
321
      0xfe 0x00 0x97 0x12 0x20 0x31 0x11 0x06 0x19 0x15 0x37 0x62 0x59 0x20 0x02 0x03 0x02
322
       `r′
            `e′
                  `c′
                        `e′
                              `i′
                                     'v'
                                          `e′
                                                `r′
                                                      `@′
                                                            `f′
                                                                  `o'
                                                                        `o'
                                                                              `.′
                                                                                    `c′
                                                                                          `o'
                                                                                                `m′
323
                                    `:′
                                          1/1
                                                \\\
                                                      ۱ť
                                                                        ۰.٬
      0x02 \h'
                   `t′
                        `t′
                               `p′
                                                            `o′
                                                                  `o'
                                                                              `c′
                                                                                    `o'
                                                                                          `m′
                                                                                                \\\
324
       `c′
             `c′
                  0x00 0x01 0x01 0x02 's'
                                                `e′
                                                      'n'
                                                            `d′
                                                                  `e′
                                                                        `r′
                                                                              `@'
                                                                                    ۱b′
                                                                                          `a′
                                                                                                `r′
325
       `c′
                                                            ۰: ۲
                                                                  \\'
                                                                        \\'
                                                                                                ۰. ′
             `o'
                   `m′
                        0x00 0x02
                                    ۱h٬
                                          `t′
                                                `t′
                                                      'p′
                                                                              ۱b′
                                                                                    `a′
                                                                                          `r′
326
       `o'
             `m′
                   \\'
                        `a′
                               `c′
                                     `c′
                                          0x00 0x01 0x01 0x08
                                                                  `n′
                                                                        `o'
                                                                              ۱
                                                                                    `e′
                                                                                          'n'
                                                                                                `c′
327
       `Υ′
                  `t′
                        ۱ì′
                                                                              ۱:'
                                                                                    \\
             `p′
                               `o′
                                     'n′
                                          0x00 0x0a `h'
                                                            ۱t'
                                                                  `t′
                                                                        `p′
                                                                                          \\'
                                                                                                `b'
328
       `r′
             `.'
                  `c′
                                     \\
                                                            0x00 0x20 0x31 0x11 0x06 0x19 0x15 0x37
                        `o′
                               `m′
                                          `a′
                                                 `c′
                                                      `c′
329
      0x62 0x59 0x20 0x03 '1'
                                     <sup>2</sup>′
                                          ۱3′
                                                ۱4′
                                                      <sup>1</sup>5′
                                                            ٬6٬
                                                                  ۲7'
                                                                        <u>، 8 '</u>
                                                                              <u>، 9</u>،
                                                                                    0x00 0x01
330
331
      The length of the original message is about 620 bytes and the encoded result is 151 bytes giving a compression ratio of
332
      about 4:1.
```

333

 0×00

`a′

۰. ′

`c′

`r′

`a′

333 2. Here is an example that covers all aspects of an envelope.

```
334
335
      <?xml version="1.0"?>
336
      <envelope>
337
        <params index="1">
338
        <to>
339
          <agent-identifier>
            <name>receiver@foo.com</name>
340
341
            <addresses>
342
              <url>http://foo.com/acc</url>
343
            </addresses>
344
            <resolvers>
345
              <agent-identifier>
346
                <name>resolver@bar.com</name>
347
                <addresses>
348
                  <url>http://bar.com/accl</url>
                  <url>http://bar.com/acc2</url>
349
350
                  <url>http://bar.com/acc3</url>
351
                </addresses>
352
              </agent-identifier>
353
            </resolvers>
354
          </agent-identifier>
355
        </to>
356
357
        <from>
358
          <aqent-identifier>
            <name>sender@bar.com</name>
359
360
            <addresses>
361
              <url>http://bar.com/acc</url>
362
            </addresses>
363
            <resolvers>
364
              <aqent-identifier>
365
                <name>resolver@foobar.com</name>
366
                <addresses>
367
                  <url>http://foobar.com/acc1</url>
368
                  <url>http://foobar.com/acc2</url>
369
                  <url>http://foobar.com/acc3</url>
370
                </addresses>
371
              </agent-identifier>
372
            </resolvers>
373
          </agent-identifier>
        </from>
374
375
376
        <comments>No comments!</comments>
377
378
        <acl-representation>fipa.acl.rep.xml.std</acl-representation>
379
380
        <payload-encoding>US-ASCII</payload-encoding>
381
382
        <date>20000508T042651481</date>
383
384
        <encrypted>no encryption</encrypted>
385
386
        <intended-receiver>
387
          <aqent-identifier>
388
            <name>intendedreceiver@foobar.com</name>
389
            <addresses>
390
              <url>http://foobar.com/acc1</url>
391
              <url>http://foobar.com/acc2</url>
392
              <url>http://foobar.com/acc3</url>
393
            </addresses>
394
            <resolvers>
395
              <agent-identifier>
396
                <name>resolver@foobar.com</name>
```

			<addr< th=""><th>resses</th><th>3></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></addr<>	resses	3>												
			<ur< th=""><th>cl>htt</th><th>p://f</th><th>Toobai</th><th>c.com/</th><th>/accl•</th><th></th><th>></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></ur<>	cl>htt	p://f	Toobai	c.com/	/accl•		>							
			<ur< th=""><th>cl>htt</th><th>p://f</th><th>Toobai</th><th>.com/</th><th>/acc2<</th><th>/url></th><th>></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></ur<>	cl>htt	p://f	Toobai	.com/	/acc2<	/url>	>							
			<ur< th=""><th>cl>htt</th><th>p://f</th><th>Toobai</th><th>c.com/</th><th>/acc3<</th><th></th><th>></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></ur<>	cl>htt	p://f	Toobai	c.com/	/acc3<		>							
			<th>iresse</th> <th>es></th> <th></th>	iresse	es>												
			<res< th=""><th>olvers</th><th>3></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></res<>	olvers	3>												
			<as< th=""><th>gent-i</th><th>denti</th><th>fier</th><th>> </th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></as<>	gent-i	denti	fier	> 										
			<	<name></name>	reso	lver@i	loobai	c.com<	<th>3></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	3>							
			<	<addre< th=""><th>sses</th><th>></th><th>,</th><th>,</th><th>1</th><th>/ 1.</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></addre<>	sses	>	,	,	1	/ 1 .							
				<ur< th=""><th>>nttp</th><th>p://IC</th><th>obar.</th><th>.com/a</th><th>acci<!--</th--><th>/url></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></th></ur<>	>nttp	p://IC	obar.	.com/a	acci </th <th>/url></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	/url>							
				<ur< th=""><th>>nttp</th><th>)://I(</th><th>obar.</th><th>. COM/a</th><th>acc2<!--</th--><th>/url></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></th></ur<>	>nttp)://I(obar.	. COM/a	acc2 </th <th>/url></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	/url>							
				ur) (add	>nttp);//I(opar.	.com/a	acc3 </th <th>/url></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	/url>							
			- / -	<th>idont</th> <th>3> -:f:o:</th> <th>~~</th> <th></th>	idont	3> -:f:o:	~~										
			20</th <th>agent-</th> <th></th> <th>e</th> <th>. /</th> <th></th>	agent-		e	. /										
		- /	'acont	idor	.5/ \+ifi4	or >											
		<th>agent</th> <th>rgy</th> <th></th> <th>=_ ></th> <th></th>	agent	rgy		=_ >											
	<	/ager	nt – i de		er>												
<	:/ir	ntend	led-re	ceive	r>												
	·, ±1	100110															
<	red	ceive	ed>														
	<1	recei	ved-k	by val	.ue="h	nttp:/	//foo.	.com/a	acc" /	/>							
	<1	recei	.ved-f	from v	value=	="http	o://fo	obar.	.com/a	acc",	/>						
	<1	recei	ved-d	late v	value=	="2000	05081	04265	51481'	" />							
	<1	recei	ved-i	id val	.ue="1	L23456	5789"	/>									
	<1	recei	ved-v	/ia va	alue='	'http:	//bai	c.com/	/acc"	/>							
<	:/re	eceiv	/ed>														
<	:/pa	arams	5>														
∈</th <th>enve</th> <th>elope</th> <th><u>></u></th> <th></th>	enve	elope	<u>></u>														
Usir	ng tl	he bit-	efficier	nt repre	sentat	ion, the	envelo	ope be	comes:	:							
0xf	le ()x01	0xea	0x12	0x20	0x31	0x11	0x06	0x19	0x15	0x37	0x62	0x59	0x20	0x02	0x02	`r′
`e′		`C′	`e′	`i′	`v′	`e′	`r′	`@′	`f′	`o <i>'</i>	`o <i>'</i>	`•′	`C′	`o <i>'</i>	`m′	0×00	0×02
۱'n		`t′	`t′	`p′	`:'	`/'	`/'	`f′	`o <i>'</i>	`o <i>'</i>	`•′	`C′	`o′	`m′	`/'	`a′	`C′
`C′	(00x	0x01	0x03	0×02	`s′	`e′	`n′	`d′	`e′	`r′	`@′	`b′	`a′	`r′	`•′	`C′
`o <i>'</i>		`m′	0x00	0×02	`h′	`t′	`t′	`p′	`:'	`/'	`/'	`b′	`a′	`r′	`•′	`C′	`o <i>'</i>
`m′		`/'	`a′	`C′	`C′	0x00	0x01	0×07	'U'	`S′	`-'	`A'	`S′	`C′	`I′	`I′	0×00
$0 \ge 0$	8	'n′	`o'	· /	`e′	`n′	`C′	`r′	`У′	`p′	`t′	`i′	`o <i>'</i>	`n′	0×00	0×01	0x09
0x0	2	`i′	`n′	`t′	`e′	`n′	`d′	`e′	`d′	`r′	`e′	`C′	`e′	`i′	`v′	`e′	`r′
`@′		`f′	`o <i>'</i>	`o <i>'</i>	`b′	`a′	`r′	`•′	`c′	`o <i>'</i>	`m′	0x00	0x02	`h′	`t′	`t′	`p′
`:'		`/'	`/'	`f′	`o <i>'</i>	`o <i>'</i>	`b′	`a′	`r′	`•′	`C′	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′
`1′	(00xC	`h′	`t′	`t′	`p′	`:'	`/'	`/'	`f′	`o <i>'</i>	`o <i>'</i>	`b′	`a′	`r′	`.′	`C′
`o <i>'</i>		`m′	`/'	`a′	`C′	`C′	`2 <i>'</i>	0x00	`h′	`t′	`t′	`p′	`:′	`/'	`/'	`f′	`o <i>'</i>
`o <i>'</i>		`b′	`a′	`r′	`.′	`C′	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′	`3′	0x00	0x01	0x03	0x02
`r′		`e′	`s′	`o′	`1′	`v′	`e′	`r′	`@′	`f′	`o <i>'</i>	`o <i>'</i>	`b′	`a′	`r'	`•′	`C′
`o′		`m′	0×00	0×02	`h′	`t′	`t′	`p′	`:'	`/'	`/'	`f′	`o′	`o <i>'</i>	`b′	`a′	`r′
`•′		`C′	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′	`1′	0x00	`h′	`t′	`t′	`q′	`:'	`/'	`/'
`f′		`o′	`o <i>'</i>	`b′	`a′	`r′	`.′	`C′	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′	`2 <i>'</i>	0x00	`h′
`t′	,	`t′	`p′	`:'	`/'	`/'	`f′	`o′	`o′	`b′	`a′	`r′	`.′	`C′	`o <i>'</i>	`m′	`/'
`a′		`C′	`C′	`3 <i>'</i>	0×00	0x01	0x03	0x02	`r′	`e′	`s′	`o <i>'</i>	`1'	`v′	`e′	`r′	`@′
`f′		`o <i>'</i>	`o′	`b′	`a′	`r′	`•′	`c′	`o′	`m′ () 00xC)x02	`h′	`t′	`t′	`p′	`:'
`/′		`/'	`f′	`o <i>'</i>	`o <i>'</i>	`b′	`a′	`r′	`.′	`C′	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′	`1 <i>'</i>
0x0	0	`h′	`t′	`t′	`p′	`:′	`/'	`/'	`f′	`o <i>'</i>	`o <i>'</i>	`b′	`a′	`r′	`.′	`C′	`o <i>'</i>
`m′		`/'	`a′	`C′	`C′	`2 <i>'</i>	0×00	`h′	`t′	`t′	`p′	`:'	`/'	`/'	`f′	`o′	`o <i>'</i>
		/	1 20 /	\ /	\a'	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′	`3 <i>'</i>	0x00	0x01	0x01	0x0a	`h′
`b′		`a'	Γ.	•	C												
`b′ `t′		`t′	'p'	`:'	`/'	`/'	`f′	`o <i>'</i>	`o <i>'</i>	`.′	`C′	`o <i>'</i>	`m′	`/'	`a′	`C′	`C′
`b' `t' 0x0		't')x20	'p' 0x31	(); (); ();	`/' 0x06	`/' 0x19	`f′ 0x15	`o′ 0x37	`o′ 0x62	`.′ 0x59	`c′ 0x20	`o' 0x02	`m′ `h′	`/' `t'	`a' `t'	`c′ `p′	`c′ `:′
`b' `t' 0x0 `/'	0 0	`t')x20 `/'	`p' 0x31 `f'	0x11 'o'	`/' 0x06 `o'	`/' 0x19 `b'	`f′ 0x15 `a′	`o' 0x37 `r'	`o′ 0x62 `.′	`.' 0x59 `c'	`c′ 0x20 `o′	`o' 0x02 `m'	`m′ `h′ `/′	`/' `t' `a'	`a' `t' `c'	`c′ `p′ `c′	`c′ `:′ 0x00
<pre>`b' `t' 0x0 `/' 0x0</pre>)0 ()3 ·	`t')x20 `/' `1'	'p' 0x31 'f' '2'	<pre>`:' Ox11 `o' `3'</pre>	`/' 0x06 `o' `4'	<pre>`/' 0x19 `b' `5'</pre>	`f′ 0x15 `a′ `6′	`o' 0x37 `r' `7'	`o' 0x62 `.' `8'	`.' 0x59 `c' `9'	`c' 0x20 `o' 0x00	`o' 0x02 `m' 0x01	`m' `h' `/' 0x01	`/' `t' `a' 0x04	`a' `t' `c' `h'	`c' `p' `c' `t'	`c' `:' 0x00 `t'

461

462 The length of the original message is about 2400 bytes and the encoded result is 490 bytes giving a compression ratio 463 of about 5:1.

464

References 4 464 465 [FIPA00067] FIPA Agent Message Transport Service Specification. Foundation for Intelligent Physical Agents, 2000. 466 http://www.fipa.org/specs/fipa00067/ 467 [FIPA00069] FIPA ACL Message Representation in Bit-Efficient Encoding Specification. Foundation for Intelligent Physical Agents, 2000. 468 469 http://www.fipa.org/specs/fipa00069/ 470 [FIPA00070] FIPA ACL Message Representation in String Specification. Foundation for Intelligent Physical Agents, 471 2000. 472 http://www.fipa.org/specs/fipa00070/ 473 [FIPA00071] FIPA ACL Message Representation in XML Specification. Foundation for Intelligent Physical Agents, 474 2000. 475 http://www.fipa.org/specs/fipa00071/