



Lightweight Message Exchange Profile (LIME)

Version 1.0.0



WP8 2009-12-22



Contents

1	Document information	4
1.1	Document history	4
1.2	Editor	4
1.3	Contributors (alphabetically).....	4
2	Introduction	5
2.1	Goals and non-goals	5
2.1.1	Goals.....	5
2.1.2	Non-Goals.....	5
2.2	Terminology.....	6
2.3	Notational conventions	6
2.3.1	Normative references	6
2.3.2	Non-normative references.....	6
2.4	Namespaces	Error! Bookmark not defined.
3	Introduction and overview	8
3.1	Example flows.....	8
4	Definition of the Message Channel	11
4.1	Concepts.....	11
4.2	BUSDOX defined headers	11
4.2.1	MessageWaiting Header	11
4.2.2	About ProcessIdentifier	11
4.2.3	About Message Identifier.....	12
4.3	Use of WS-Transfer.....	12
4.3.1	Securing channels	12
4.3.2	Use of WS-Addressing Reference Parameters	13
4.4	Inbound Message Channel	13
4.4.1	Finding available messages	14
4.4.2	Getting a message using WS-Transfer	18
4.4.3	Inclusion of SAML attributes.....	18
4.4.4	Deleting messages.....	18

4.5	The Outbound Message Channel	18
4.6	Message Sending	19
4.6.1	Faults	21
4.7	Use of HTTP	22
4.8	Use of MTOM	22
5	Security	22
Appendix C – XML Schema for Lime Types		23

1 Document information

2 1.1 Document history

Date	Version	Initials	Changes
2009-01-03	0.1.0	GS	Created template
2009-02-06	0.1.1	PZF	Initial definition and thoughts
2009-02-17	0.3	PZF	Updates based on feedback from GS Significantly changed to be completely WSTransfer based Renamed store to channel
2009-02-17	0.5	GS	Minor editorial changes
2009-03-29	0.6	PZF	Updates based on the 0.5 feedback
2009-04-01	0.7	GS	Minor editorial updates
2009-04-30	0.8	PZF	Restructuring and significant updates based on 0.7 feedback meeting
2009-08-30	0.9	PZF	Updates based on F2F meeting in Copenhagen plus feedback
2009-09-01	0.9	GS	Updated schema, naming + namespaces, references, example XML
2009-09-08	0.9.1	GS	Changed attribute "numOfEntries" to "numberOfEntries" Changed sect. 3.4 "Inbound Message Channel" to include 'EntryList'
2009-09-09	0.9.1	GS	Changed 'type' attribute of most identifiers to 'scheme'
2009-10-22	0.9.5	GS	Updated example XML to valid instance, added example of 'NextPageIdentifier' element, changed size of @size to 'long', added minOccurs to 'NextPageIdentifier' element, removed whitespace from TNS in examples, wsa:EndpointReference is now child element of 'NextPageIdentifier'.
2009-11-17	1.0.0	PZF	Final updates for 1.0.0
2009-11-27	1.0.0	GS	Added synchronous delivery fault, moved section on message headers to the 'common definitions' document, described how documents that are not part of a process are handled, and minor editorial changes.
2009-12-22	1.0.0	PZF	Removed MessageUndelivered (into Common Definitions) and added fixes based on Philip Helger's feedback

3

4 1.2 Editor

5 Paul Fremantle, WSO2

6 1.3 Contributors (alphabetically)

7 Jens Jakob Andersen, NITA

8 Mikkel Hippe Brun, NITA

9 Mike Edwards, IBM

10 Paul Fremantle, WSO2

11 Thomas Gundel, IT Crew

12 Philip Helger, Bundesrechenzentrum

13 Hans Guldager Knudsen, Lenio

14 Bergthór Skúlason, NITA

15 Gert Sylvest, Avanade

16 **2 Introduction**

17 The Lightweight Message Exchange Profile (LIME) provides a simple low-cost approach for Small and
18 Medium Enterprises (SMEs) to access Business Document Exchange Network (BUSDOX) infrastructure. The
19 “low costs” that this profile is designed to address includes:

- 20 • No requirement to host online endpoints, hence no firewall crossing, no server infrastructure.
- 21 • No requirement to support “advanced” WS-* standards such as WS-Trust, WS-ReliableMessaging. Only
22 minimal requirement to support WS-Addressing and WS-Transfer only. Since WS-Transfer is a simple
23 WSDL-based specification, the only requirement on the SOAP stack is to support WS-Addressing.

24 This is achieved through the use of a Business Document Exchange Network (BUSDOX) Access Point [BDEN-
25 CDEF] that supports this profile and manages messages on behalf of the client. It both handles messages
26 destined for the client by storing them in a Message Channel awaiting retrieval and also the Relay Service
27 provides a simple way that the client may send messages to other organizations without requiring to
28 navigate the service metadata. A simple analogy is the POP3/SMTP-Relay services that ISPs provide that
29 enables email access from intermittently connected computers.

30 **2.1 Goals and non-goals**

31 **2.1.1 Goals**

- 32 • Provide an interface to a message channel and relay service that supports intermittently connected
33 systems.
- 34 • Provide access over a simple HTTPS-protected channel
- 35 • Utilize existing standards where appropriate
- 36 • Support the same message format as other BUSDOX Transport Profiles
- 37 • Lower the cost of entry for SMEs and individuals.

38 **2.1.2 Non-Goals**

- 39 • This profile does not support end-to-end security or identity. The BUSDOX Lightweight Message
40 Exchange Profile Access Point (LIME-AP) must validate the credentials of customers using the LIME
41 profile and map those credentials into a valid identity to be used for outbound communications.
- 42 • This specification is expected to be used in the context of a particular usage of the BUSDOX profiles: for
43 example, the types and formats of participant identifiers are not specified as part of this profile, but in
44 a real deployment would be specified as part of a governance model.

45 **2.2 Terminology**

46 Please see Common Definitions [BDEN-CDEF] section 2.2

47 **2.3 Notational conventions**

48 Notational conventions have been adopted from [WSDL-2.0], see "Common Definitions" [BDEN-CDEF]
49 section 2.2.

50 Pseudo-schemas are provided for each component, before the description of the component. They use
51 BNF-style conventions for attributes and elements: "?" denotes optionality (i.e. zero or one occurrences),
52 "*" denotes zero or more occurrences, "+" one or more occurrences, "[" and "]" are used to form groups,
53 and "|" represents choice. Attributes are conventionally assigned a value which corresponds to their type,
54 as defined in the normative schema. Elements with simple content are conventionally assigned a value
55 which corresponds to the type of their content, as defined in the normative schema. Pseudo schemas do
56 not include extension points for brevity.

```
57  
58 <!-- sample pseudo-schema -->  
59 <defined_element  
60     required_attribute_of_type_string="xs:string"  
61     optional_attribute_of_type_int="xs:int"? >  
62 <required_element />  
63 <optional_element />?  
64 <one_or_more_of_these_elements />+  
65 [ <choice_1 /> | <choice_2 /> ]*  
66 </defined_element>
```

67 **2.3.1 Normative references**

68 [BDEN-CDEF] Business Document Exchange Network - Common Definitions, CommonDefinitions.pdf

69 [WS-T] "Web Services Transfer (WS-Transfer)", W3C Working Draft 24 September 2009,
70 <http://www.w3.org/TR/2009/WD-ws-transfer-20090924/>

71 [WSA-1.0] "Web Services Addressing 1.0 - Core" (<http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/>) and "Web Services Addressing 1.0 - SOAP Binding", <http://www.w3.org/TR/ws-addr-soap/>

73 [XML-DSIG] "XML Signature Syntax and Processing (Second Edition)", <http://www.w3.org/TR/xmlsig-core/>

74 [RFC-2119] "Key words for use in RFCs to Indicate Requirement Levels", <http://www.ietf.org/rfc/rfc2119.txt>

75 [SOAP-1.1] "Simple Object Access Protocol (SOAP) 1.1", <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

77 **2.3.2 Non-normative references**

78 [WSDL-2.0] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language",
79 <http://www.w3.org/TR/wsdl20/>

80 **2.4 Namespaces**

81 The following table lists XML namespaces that are used in this specification. The choice of any namespace
82 prefix is arbitrary and not semantically significant.

Namespace Prefix	Namespace
wsa	http://www.w3.org/2005/08/addressing
s	http://schemas.xmlsoap.org/soap/envelope/
lime	http://busdox.org/transport/lime/1.0/
ids	http://busdox.org/transport/identifiers/1.0/
xs	http://www.w3.org/2001/XMLSchema

83

84

85 **3 Introduction and overview**

86 **3.1 Example flows**

87 The Lightweight Message Exchange Profile is designed to allow systems to participate in the BUSDOX
88 infrastructure without needing to access service metadata directly or host an Access Point. Instead, they
89 may choose to use the Lightweight Message Exchange Profile (LIME) to communicate with a service
90 provider. A simple analogy is Internet email: Large companies may run their own Simple Mail Transport
91 Protocol (SMTP) server and proprietary email clients to create and read messages, but individuals or small
92 companies rely on an ISP to provide an SMTP Relay and POP3 or IMAP server.

93 This profile describes the approach that a LIME Client (LC) can use to send and receive messages from an
94 LIME-enabled Access Point.

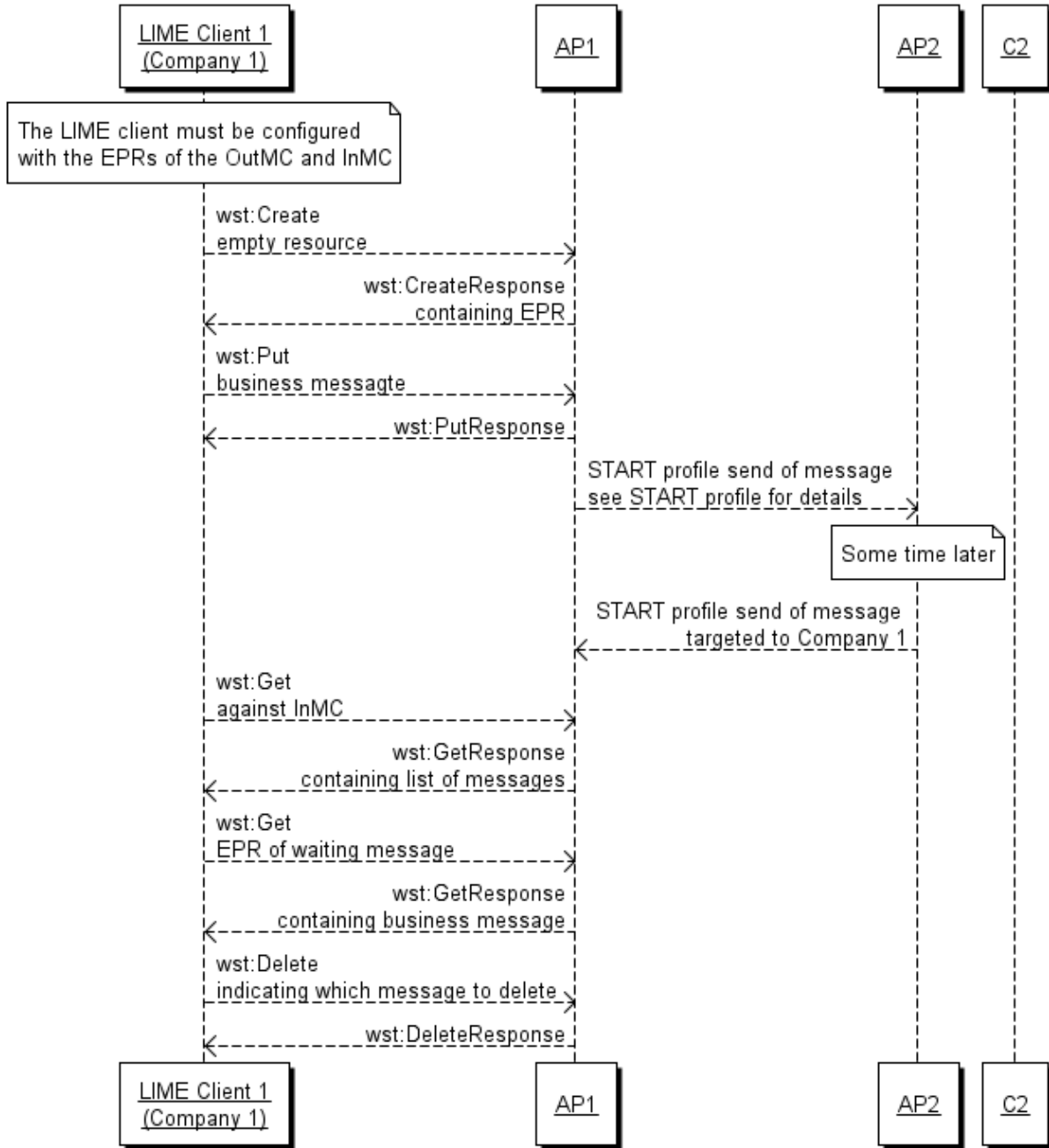
95 The diagram below shows a simple example flow. The LC needs to send a message to a company which
96 uses an Access Point we will call AP2. However, the LC only needs to be configured to talk to a single local
97 access point (AP). Initially the business user creates a business message using the software (out of scope for
98 this profile). The requirements are that the business message complies with the BUSDOX specifications and
99 that the correct participant identifiers (see section 4.2) are made available to the LC.

100 The LC sends a Create message to the AP, which initiates the message flow and causes the AP to create a
101 fixed Endpoint Reference (EPR) Resource. The message is then put into this resource by the LC. This model
102 ensures that messages delivered exactly once to the AP. Once the message is delivered to the access point,
103 it looks up the recipient's AP and transfers the message.

104

105 The LC also polls the AP for any incoming messages. This is done by “Get”-ting a list of available messages
 106 from the AP, and then individually retrieving each available message (if any) using another Get.

107



108

109

Figure 1 - Sequence diagram showing a typical LIME profile usage

110 3.2 Technical Overview of the Profile

111 The profile defines a set of technologies that are used together:

- 112 • HTTPS and Basic Authentication for security

- 113 • SOAP 1.1 for the base communications
 - 114 • WS-Transfer as a standard approach to accessing the message channels
 - 115 • BUSDOX specific headers to define standard metadata
 - 116 • BUSDOX specific XML Schema to define the message list XML format
- 117 Together these different technologies are used together to define a simple protocol that can allow an
118 intermittently connected computer to fully participate in a BUSDOX infrastructure so long as they have
119 a Lightweight Message Exchange Profile Access Point (LIME-AP) available.
- 120

121 **4 Definition of the Message Channel**

122 **4.1 Concepts**

123 A message channel is a WS-Transfer endpoint that either accepts or retrieves messages from an LC. A single
124 channel may handle both incoming and outgoing messages or there may be independent channels. The
125 profile assumes that there may be independent channels and therefore that the LC is provided with
126 addressing information for both channels. For the remainder of this specification all references assume that
127 there are two independent channels. However, the specification is written in such a way that there may be
128 a single channel - in either case the specification can operate correctly.

129 The Outbound message channel (OutMC) accepts outbound messages (messages from LC to AP) using the
130 WS-Transfer Create and Put operations, and the inbound message channel (InMC) offers inbound messages
131 (messages from AP to LC) using the WS-Transfer Get method, and allows these to be deleted using the WS-
132 Transfer Delete method.

133 **4.2 BUSDOX defined headers**

134 Every BUSDOX message has associated metadata included so that Access Points can route messages
135 without needing to look inside the business message. Therefore this profile defines the following
136 mandatory header blocks.

137 The Common Definitions document [BDEN-CDEF] defines the following identifiers in section 3.7:

- 138 • RecipientIdentifier
- 139 • SenderIdentifier
- 140 • DocumentIdentifier
- 141 • ProcessIdentifier
- 142 • MessageIdentifier
- 143 • ChannelIdentifier

144 For an XML Schema for these elements, see 'Common Definitions' [BDEN-CDEF].

145 **4.2.1 MessageWaiting Header**

146 The LIME-enabled Access Point MAY indicate to a LIME client that there are messages waiting in the
147 Outbound Message Channel. This header MAY be added into any response message flowing to the LIME
148 Client. The header element is:

```
149 | <lime:MessageWaiting/>
```

150 The header MUST NOT have any attributes.

151 **4.2.2 About ProcessIdentifier**

152 If a document is not part of a well defined business process, the ProcessIdentifier header must still be
153 present. It then MUST hold the value 'busdox:noprocess' and scheme 'busdox-procid-transport', see the
154 section on process identifiers in [BDEN-CDEF].

155 **4.2.3 About Message Identifier**

156 Because BUSDOX Messages may pass between several parties (for example in the “four-corner” model,
157 from LC to AP to AP to LC), it is necessary to have a constant message identifier that uniquely identifies the
158 message across multiple hops. This message identifier is contained in the:

```
159 <ids:MessageIdentifier>  
160 element.
```

161 During message resending in the PUT phase, the MessageIdentifier header MUST be the same for each
162 resend of the same message. The ids:MessageIdentifier is created by the AP (as a reference parameter),
163 and is then sent along with the business message as it passes on to other APs.

164 **4.3 Use of WS-Transfer**

165 For access to the Message Channels in this profile the LC uses the WS-Transfer specification [BDEN-CDEF]
166 Business Document Exchange Network - Common Definitions, CommonDefinitions.pdf

167 [WS-T]¹. WS-Transfer is used to send messages from the LC to the AP as well as retrieve waiting messages.

168 In order to ensure the reliable sending of messages we use a pattern that we call CreatePut².

169 Receiving messages is done with two or more Gets – the first lists a page of available messages, further
170 requests may retrieve individual messages or further pages of message listing. Messages SHOULD be
171 deleted (using WS-T Delete) once successfully retrieved. WS-Transfer does not define a resource listing
172 model, so this profile defines a simple XML Schema for lists.

173 **4.3.1 Securing channels**

174 The LIME-AP MAY secure the two message channels in the following fashion:

- 175 • The LC can list messages using the Get interface on the InMC. The LC can Get and Delete messages
176 in the Inbound Message Channel. It cannot Create or send (Put) messages in the Inbound Message
177 Channel.
- 178 • The LC may Create and send (Put) messages in the Outbound Message Channel. It cannot list
179 messages, Get, or Delete messages in the OutMC.

180 Of course, this model assumes that there are two independent channels. In the case where there is a single
181 channel operating as both InMC and OutMC, the associated security must be determined by the AP.

¹ It is expected that future versions of LIME or errata will update to the Final version of WS-Transfer as and when this becomes available.

² This is based on a common pattern used with REST services.

182 In this model it is possible to have a single outbound message channel shared by many companies. This is
183 not normative. Another possible alternative is that the same channel identifier is used to both send and
184 receive messages.

185 The LIME-AP MUST support the listing interface and WS-Transfer GET/DELETE on the InMC.

186 The LIME-AP MUST support CREATE/PUT on the OutMC.

187 **4.3.2 Use of WS-Addressing Reference Parameters**

188 WS-Transfer supports the use of any WS-Addressing Reference Parameters to define resources that are
189 transferred. However, for the purpose of this profile, we define specific SOAP headers/reference
190 parameters to be used. These headers MUST be used. The profile authors understand the W3C guidance
191 that EPRs are designed to be opaque. However, the authors believe there are two significant benefits to
192 specifying the reference parameters:

- 193 • A clear basis for comparing endpoint references, since EPRs are clearly defined.
- 194 • The configuration of the LC is simpler, because channels can always be configured with a
195 combination of URL and Channel Identifier.

196 The ChannelIdentifier is a URI which uniquely identifies a channel. Every WS-Transfer request against a
197 channel MUST have the ChannelIdentifier reference parameter present.

```
198 <ids:ChannelIdentifier>xs:anyURI</ids:ChannelIdentifier>
```

199

200 The MessageIdentifier is a URI which uniquely identifies a message. The message identifier is consistent
201 across multiple hops.

```
202 <ids:MessageIdentifier>xs:anyURI</ids:MessageIdentifier>
```

203 **4.4 Inbound Message Channel**

204 The LC retrieves messages from a specific Inbound Message Channel (InMC), identified by an Endpoint
205 Reference (EPR) provided by the LIME-AP. The EPR contains a unique identifier for the Channel known as
206 the Channel Identifier (ChannelIdentifier). For example the EPR of the Inbound Message Channel may
207 contain a Channel Identifier (ChannelIdentifier) that is based on the company registration number. Please
208 note that the actual ChannelIdentifier is defined by the AP's system and is only relevant when talking to
209 that access point.

210 Here is an example EPR for an inbound message channel:

```
211 <wsa:EndpointReference>  
212   <wsa:Address>  
213   http://LIME-AP.my-van.com:80/services/messagechannel  
214   </wsa:Address>  
215   <wsa:ReferenceParameters>  
216     <ids:ChannelIdentifier>55038353</ids:ChannelIdentifier>  
217   </wsa:ReferenceParameters>  
218 </wsa:EndpointReference>
```

219 The manner in which this EPR is provided to the LC is out-of-scope: for example it may be manually entered
220 as part of the user configuration of the LC.

221 The LC may have many message channels that it can access. The Message Channel may store any number
222 of messages.

223 In order to allow the LC to find and access these messages a three-step process is used:

- 224 1. First the LC uses the WS-Transfer Get operation to retrieve a list of messages that are waiting to in the
225 channel.
- 226 2. The LC uses the WS-Transfer interface on the EPR to retrieve (GET) the message. If there is a failure
227 retrieving the message, the LC may repeat this step as needed.
- 228 3. Once the message is successfully retrieved, the LC SHOULD use the WS-Transfer Delete operation to
229 delete the message from the channel.

230 An example flow can be seen in Figure 1.

231 Each individual message in a channel has an Endpoint Reference which contains both the ChannelIdentifier
232 as well as a unique MessageIdentifier as reference parameters. Here is an example of an Endpoint
233 Reference for a message.

```
234 <wsa:EndpointReference>  
235   <wsa:Address>  
236   http://LIME-AP.van.co.uk:80/services/transfer  
237   </wsa:Address>  
238   <wsa:ReferenceParameters>  
239     <ids:ChannelIdentifier>55038353</ids:ChannelIdentifier>  
240     <ids:MessageIdentifier>  
241       uuid:f8290-432lkj2349-8aiuyfga0  
242     </ids:MessageIdentifier>  
243   </wsa:ReferenceParameters>  
244 </wsa:EndpointReference>
```

245 **4.4.1 Finding available messages**

246 To find available messages, the LC simply does a WS-Transfer Get on the Channel – in other words the Get
247 is targeted against the channel EPR.

248 If the channel access control permissions allow this, the channel responds with an XML list of available
249 messages. In order to be efficient, the list can be paginated. The Channel decides a maximum number of
250 messages to list in a page, and each page of the list is a resource that can be transferred. Performing a GET
251 on the main channel EPR always returns start of the list. The list is ordered by the time of creation of the
252 messages.

253 The LC may adopt two strategies for downloading messages:

- 254 • The LC may simply retrieve the first page, download all listed messages, delete them, and then
255 relist. The next page will then be listed.

- 256 • The LC may retrieve the first page of listed messages. At the end of this list is the EPR of the next
257 page. The LC may then retrieve the next page, and so forth until it has listed all available messages.

258 The full schema for the paginated list interface is in the Appendix, and the definition of the meaning of the
259 XML document elements is defined as follows.

260 The operation provides a paginated list of available messages. Each page has up to n messages, where n is a
261 number defined by the channel server system. It is recommended that n=100 as a simple default.

262 **/lime:PageList**

263 This element contains a page of entries, which may be downloaded individually. It also
264 contains a reference to additional pages of entries, if such exist.

265 **/lime:PageList /lime:EntryList**

266 This element contains the individual entries; each entry is a document that can be downloaded
267 by the client.

268 **/lime:PageList/lime:EntryList /@numberOfEntries**

269 This attribute is a long number, containing the number of entries returned in this page. This
270 number will match the number of /lime:PageList/lime:Entry elements that are children of this
271 /lime:Pagelist element.

272 **/lime:PageList/lime:EntryList /lime:Entry**

273 This element contains information about a specific document that is waiting for collection.

274 **/lime:PageList/lime:EntryList/lime:Entry/@size**

275 This required attribute captures the size of the document waiting for collection, in Kilobytes
276 (multiples of 1024), as a rounded long integer numeric value.

277 **/lime:PageList/ lime:EntryList /lime:Entry/@creationTime**

278 This required attribute whose value is an XML Schema dateTime captures the time that the
279 document referenced by this Entry was stored by the access point.

280 **/lime:PageList/ lime:EntryList /lime:Entry/@messageBodyLocalName**

281 This optional attribute captures the tag name of the first element of the document referred to
282 by the Entry. In the case that the document is encrypted this will not be available to the Access
283 Point and this attribute will not be present.

284 **/lime:PageList/ lime:EntryList /lime:Entry/@messageBodyNamespace**

285 This optional attribute (type xs:anyURI) captures the namespace of the first element of the
286 document referred to by the Entry. In the case that the document is encrypted this will not be
287 available to the Access point and this attribute will not be present.

288 The EPR for listing messages from a channel SHOULD contain the Message Channel Identifier
289 (ChannelIdentifier), and the EPR Reference Parameters MUST be included in the SOAP Header
290 of any request messages.

291 Any EPRs offered by the channel for *listing* messages MUST NOT include the Reference Parameter
292 <ids:MessageIdentifier>

293

294 Example:

```
295 <ids:ChannelIdentifier wsa:isReferenceParameter="true">
296     Channel Identifier
297 </ids:ChannelIdentifier>
298 <lime:PageIdentifier wsa:isReferenceParameter="true">
299     5
300 </lime:PageIdentifier>
301
```

302 The WS-Transfer page list XML Schema is in the appendix.

303 For the purposes of this profile, the EndpointReferences returned in the sequence of Entries MUST contain
304 the following two reference parameters:

```
305 <ids:ChannelIdentifier>xs:string</ids:ChannelIdentifier>
306 <ids:MessageIdentifier>xs:string</ids:MessageIdentifier>
307
```

308 Here is a sample XML response to the page listing GET request:

```
309 <?xml version="1.0" encoding="utf-8" ?>
310 <!--
311     An sample XML response to the page listing GET request
312 -->
313 <lime:PageList
314     xmlns:lime="http://busdox.org/transport/lime/1.0/"
315     xmlns:wsa="http://www.w3.org/2005/08/addressing"
316     xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
317     numberOfEntries="1">
318     <lime:EntryList>
319         <lime:Entry size="8295" creationTime="2009-02-18T12:33:45Z"
320             messageBodyLocalName="Order"
321             messageBodyNamespace="http://busdox.org/ns/Order">
322             <wsa:EndpointReference>
323                 <wsa:Address>
324                     http://LIME-AP.my-van.com:80/services/transfer
325                 </wsa:Address>
326                 <wsa:ReferenceParameters>
327                     <ids:ChannelIdentifier>55082098</ids:ChannelIdentifier>
328                     <ids:MessageIdentifier>uuid:45989-2429-
329 132412312</ids:MessageIdentifier>
330                 </wsa:ReferenceParameters>
331             </wsa:EndpointReference>
332         </lime:Entry>
333     </lime:EntryList>
334
335     <lime:NextPageIdentifier>
336         <wsa:EndpointReference>
337             <wsa:Address>
338                 http://LIME-AP.my-van.com:80/services/messagechannel
339             </wsa:Address>
340             <wsa:ReferenceParameters>
341                 <ids:ChannelIdentifier>55038353</ids:ChannelIdentifier>
342             </wsa:ReferenceParameters>
343         </wsa:EndpointReference>
344     </lime:NextPageIdentifier>
345 </lime:PageList>
346
347     NOTE: The 'PageIdentifier' may be replaced by element in
```

```

345         any namespace that represents a system-specific ID of the next
346 page
347     -->
348     <PageIdentifier
349 xmlns="http://someNamespace.org">2</PageIdentifier>
350     </wsa:ReferenceParameters>
351     </wsa:EndpointReference>
352 </lime:NextPageIdentifier>
353
354
355 </lime:PageList>
356

```

357 The LC MUST use document/literal binding to access the Channel WS-Transfer service.

358 If the message was transferred into the channel using a BUSDOX Transport profile, then the
359 MessageIdentifier used as a reference parameter MUST be the same as the ids:MessageIdentifier of the
360 message used to create the message in the channel. If no such MessageIdentifier exists, then the LIME-AP
361 should create a guaranteed unique MessageIdentifier for the message.

362 4.4.2 Getting a message using WS-Transfer

363 Once an Endpoint Reference has been retrieved using the Get listing operation, the message may be
364 retrieved using the WS-Transfer Get method.

365 All BUSDOX defined headers that were transferred to the LIME-AP MUST be included as SOAP Headers
366 when the message is retrieved using Get.

367 4.4.3 Inclusion of SAML attributes

368 If the message being retrieved from the channel originated in another BUSDOX access point, then it will
369 have had a SAML token attached at that point with an assurance level attribute. In order to support end-to-
370 end traceability and assurance, the LIME-AP MUST include assurance level attribute in any messages that
371 are made available in the inbound channel.

372 The following header contains the SAML attribute:

```

373 <lime:identityAssurance>
374   <saml2:Attribute/>
375 </lime:identityAssurance>

```

376 The LC MUST include the <saml2:Attribute Name="urn:eu:busdox:attribute:assurance-level" > element. The
377 LC MAY use this information to inform the business users of the BUSDOX assurance level.

378 4.4.4 Deleting messages

379 It is the responsibility of the LC to delete messages once they have been retrieved. The WS-Transfer DELETE
380 operation SHOULD be used. Both reference parameters (ChannelIdentifier and MessageIdentifier) MUST be
381 used to delete messages.

382 In the case there is an error during the Delete (for example a dropped connection) it is the responsibility of
383 the LIME client to retry until there is a confirmation that the resource no longer exists.

384 4.5 The Outbound Message Channel

385 The Outbound Message Channel (OutMC) provides a simple model where the LC may transfer messages to
386 a LIME-AP. These messages are then transferred to other BUSDOX Access Points using business addressing
387 information stored in the business message.

388 **4.6 Message Sending**

389 In this exchange, the LC and the LIME-AP implement a reliable delivery model to ensure that messages are
390 delivered once-only. This is known as the CreatePut model.

391 In order to implement a simple reliable idempotent model for relaying messages outbound, the LC
392 implements a two-stage message sending process:

- 393 1. In the first stage, the LC creates an *empty* resource in the OutMC, not containing the real
394 message. This is done using the CREATE request and with no business message
395 `<wst:Create/>`
396 The response to this is:
397 `<wxf:ResourceCreated>endpoint-reference</wxf:ResourceCreated>`
398 which contains an EndpointReference of the resource that will be used to transmit the message.
399 The EndpointReference will contain a unique `ids:MessageIdentifier` reference parameter.
400 If this operation is incomplete or the response is dropped at the network level, the LC should
401 retry. In this case, there may be an extra unused Resource and EPR available on the LIME-AP
402 (from the first failed CREATE request). The LIME-AP SHOULD keep such resources available for up
403 to one hour, to allow for timing issues in the LC. The LIME-AP SHOULD garbage collect/delete any
404 such resources that remain empty for extended periods of time.

405
406 The Create message MUST include the following BUSDOX defined headers in the SOAP Header:
407 *ids:RecipientIdentifier, ids:ChannelIdentifier, ids:SenderIdentifier, ids:DocumentIdentifier,*
408 *ids:ProcessIdentifier*

409
410 These headers MUST be used by the OutMC when sending this message onwards.

411
412 In addition the `ids:MessageIdentifier` header will be included as one of the reference parameters.

413
414 The client MAY include other headers in the message. Any headers that are not defined in the
415 BUSDOX-namespace and are not part of the Reference Parameters SHOULD be stored and
416 SHOULD be relayed onward with the business message.

- 417 2. In the second stage the LC uses the WS-Transfer Put operation to transfer the actual message to
418 the EPR returned in stage 1.

419
420 The Put message request SHOULD NOT include the BUSDOX defined headers in the SOAP Header,
421 except the reference parameters. Other non-BUSDOX headers SHOULD NOT be included. Any
422 headers that are not defined in the BUSDOX-namespace and are not part of the Reference
423 Parameters SHOULD be dropped by the Access Point before transferring to another Access Point.
424

425 The LC SHOULD repeat this second step as often as required until it gets a successful response. If
426 there is a long period of time between step 1 and step 2, it is possible that the LIME-AP has
427 deleted the resource. In this case the LIME-AP will return a fault to the LC indicating an unknown
428 EPR. In this case, the LC SHOULD restart at step 1.

429
430 The LC SHOULD log the PUT request message and PUT response message for proof-of-delivery.
431 The LIME-AP MUST include the WS-Addressing RelatesTo header.

432
433 This profile explicitly defines the format of the EndpointReferences used to Create resources in
434 the Outbound Message Channel. The <wsa:MessageID> from the Initial Create message MUST be
435 used to create the EndpointReference in the following way:

```
436 <wsa:EndpointReference>  
437   <wsa:Address>  
438     http://LIME-AP.my-van.com:80/services/messagechannel  
439   </wsa:Address>  
440   <wsa:ReferenceParameters>  
441     <ids:ChannelIdentifier>outbound</ids:ChannelIdentifier>  
442   </wsa:ReferenceParameters>  
443 </wsa:EndpointReference>
```

444

445 When the Channel returns a new endpoint reference from the Create operation, it adds a unique
446 MessageIdentifier to the ReferenceParameters, e.g.:

```
447 <wxf:ResourceCreated>  
448   <wsa:EndpointReference>  
449     <wsa:Address>  
450       http://LIME-AP.my-van.com:80/services/messagechannel  
451     </wsa:Address>  
452     <wsa:ReferenceParameters>  
453       <ids:ChannelIdentifier>outbound</ids:ChannelIdentifier>  
454       <ids:MessageIdentifier>uuid:45989-2429-  
455 132412313</ids:MessageIdentifier>  
456     </wsa:ReferenceParameters>  
457   </wsa:EndpointReference>  
458 </wxf:ResourceCreated>
```

459

460 Once the LIME-AP receives a complete successful Put operation it can relay the message on to the final
461 recipient. The LIME-AP should treat subsequent Puts of the same message as correct, as these indicate that
462 the LC has not yet received a successful response and will keep retrying. The LIME-AP SHOULD use the
463 <ids:MessageIdentifier> of the Put request when relaying messages onward.

464 The LIME-AP MUST generate unique Message IDs for the endpoint reference returned in the
465 CreateResponse.

466 The LIME-AP MUST NOT attempt to deliver empty messages.

467

468 **4.6.1 Faults**

469 The DestAP can fault in four circumstances on the OutMC. Firstly, it may have a “full channel”. This
 470 indicates that the client should retry at a later time. Secondly, the endpoint may not be recognized. Thirdly,
 471 there may be a security error. Finally, there may be an internal server fault (Server Error). The faults used
 472 are as follows:

473 **Channel Full Fault**

[action]	http://busdox.org/2010/02/channel/fault
Code	s:Sender
Subcode	lime:ChannelFull
Reason	The channel is not accepting messages for this destination
Detail	As detailed by the AP

474 **Unknown Endpoint**

[action]	http://busdox.org/2010/02/channel/fault
Code	s:Sender
Subcode	lime:UnknownEndpoint
Reason	The endpoint is not known
Detail	As detailed by the AP

475 **Security Error**

[action]	http://busdox.org/2010/02/channel/fault
Code	s:Sender
Subcode	lime:SecurityFault
Reason	There is a security error in processing this request
Detail	As detailed by the AP

476 **Server Error**

[action]	http://busdox.org/2010/02/channel/fault
Code	s:Sender

Subcode	lime:ServerError
Reason	ServerError
Detail	As detailed by the AP

477 **4.7 Use of HTTP**

478 Please see the Common Definitions document for use of HTTP [BDEN-CDEF].

479 **4.8 Use of MTOM**

480 The Message Transmission Optimization Mechanism is a way of effectively encoding binary data in SOAP
481 messages. LIME Clients MAY use MTOM to send messages. If an LC supports MTOM, it MAY use an MTOM
482 packaging to issue a WS-Transfer GET request. In this case the LIME-AP MUST respond with an MTOM
483 encapsulated message. The LIME-AP MUST support MTOM on the LIME services.

484 **5 Security**

485 It is up to the LIME-AP to manage the user access to channels, based on the HTTP Basic Authentication or
486 other authentication credentials provided to the LIME-AP by the LC.

487 It is important to note that the security of the Lightweight Message Exchange Profile is only point-to-point
488 and not end-to-end. This means that the credentials used to authenticate the LC to the LIME-AP need not
489 be acceptable by other BUSDOX Access Points. The credentials are only required to be accepted by the
490 LIME-AP. For example, the LIME-AP may run its own user channel for small companies and map these
491 credentials into tokens acceptable by other BUSDOX Access Points.

492 The minimum required security for LIME is to use:

- 493 • HTTP Basic authentication
- 494 • Transport Layer Security for encryption

495

496 **Appendix C– XML Schema for Lime Types**

497 XSD for the Lime Types:

```
498 <?xml version="1.0" encoding="UTF-8" ?>
499 <schema
500     targetNamespace="http://busdox.org/transport/lime/1.0/"
501     elementFormDefault="qualified"
502     xmlns="http://www.w3.org/2001/XMLSchema"
503     xmlns:tns="http://busdox.org/transport/lime/1.0/"
504     xmlns:wsa="http://www.w3.org/2005/08/addressing"
505     xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
506     version="1.0.0">
507
508     <import namespace="http://www.w3.org/2005/08/addressing"
509 schemaLocation="ws-addr.xsd" />
510     <import schemaLocation="Identifiers-1.0.xsd"
511 namespace="http://busdox.org/transport/identifiers/1.0/" />
512
513     <element name="MessageUndeliverable"
514 type="tns:MessageUndeliverableType" />
515
516     <complexType name="MessageUndeliverableType">
517         <sequence>
518             <element ref="ids:MessageIdentifier" />
519             <element name="ReasonCode" type="tns:ReasonCodeType" />
520             <element name="Details" type="string" />
521         </sequence>
522     </complexType>
523
524     <simpleType name="ReasonCodeType">
525         <restriction base="string">
526             <enumeration value="METADATA_ERROR" />
527             <enumeration value="TRANSPORT_ERROR" />
528             <enumeration value="SECURITY_ERROR" />
529             <enumeration value="OTHER_ERROR" />
530         </restriction>
531     </simpleType>
532
533     <element name="PageList" type="tns:PageListType" />
534
535     <complexType name="PageListType">
536         <sequence>
537             <element name="EntryList" type="tns:EntryListType" />
538             <element name="NextPageIdentifier"
539 type="tns:NextPageIdentifierType" minOccurs="0" />
540         </sequence>
541         <attribute name="numberOfEntries" type="long"
542 use="optional"></attribute>
543     </complexType>
544
545     <complexType name="EntryListType">
546         <sequence>
```

```
547         <element name="Entry" type="tns:Entry" minOccurs="0"
548 maxOccurs="unbounded" />
549     </sequence>
550 </complexType>
551
552 <complexType name="Entry">
553     <sequence>
554         <element ref="wsa:EndpointReference" />
555     </sequence>
556     <attribute name="size" type="long"></attribute>
557     <attribute name="creationTime" type="dateTime"
558 use="optional"></attribute>
559     <attribute name="messageBodyLocalName" type="string"
560 use="optional"></attribute>
561     <attribute name="messageBodyNamespace" type="anyURI"
562 use="optional"></attribute>
563 </complexType>
564
565 <complexType name="NextPageIdentifierType">
566     <sequence>
567         <element ref="wsa:EndpointReference" />
568     </sequence>
569 </complexType>
570 </schema>
571
```