



# Lightweight Message Exchange Profile (LIME)

---

*Version 0.9.5*



WP8 2009-10-22



## Contents

1	Document information .....	4
1.1	Document history .....	4
1.2	Editor .....	4
1.3	Contributors (alphabetically).....	4
	Introduction.....	5
1.4	Goals and non-goals .....	5
1.4.1	Goals .....	5
1.4.2	Non-Goals .....	5
1.5	Terminology.....	5
1.6	Notational conventions .....	5
1.6.1	Normative references.....	6
1.6.2	Non-normative references .....	6
1.7	Namespaces.....	7
2	Introduction and overview .....	8
2.1	Example flows.....	8
2.2	Technical Overview of the Profile.....	10
3	Definition of the Message Channel .....	11
3.1	Concepts .....	11
3.2	BUSDOX defined headers .....	11
3.2.1	Recipient Business Identifier .....	11
3.2.2	Sender Business Identifier .....	11
3.2.3	Document Type Identifier.....	11
3.2.4	Process Type Identifier .....	12
3.2.5	Message Identifier .....	12
3.3	Use of WS-Transfer.....	12
3.3.1	Securing channels .....	12
3.3.2	Use of WS-Addressing Reference Parameters .....	13
3.4	Inbound Message Channel .....	13
3.4.1	Finding available messages.....	14

3.4.2	Getting a message using WS-Transfer .....	18
3.4.3	Inclusion of SAML attributes .....	18
3.4.4	Deleting messages .....	18
3.5	The Outbound Message Channel .....	18
3.6	Message Sending .....	18
3.6.1	Faults .....	20
3.7	Use of HTTP .....	22
3.8	Use of MTOM .....	22
4	Security .....	22
5	Appendix B – XML Schema for headers.....	23
6	Appendix C – XML Schema for PageList XML .....	24

# 1 Document information

## 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'.

## 1.2 Editor

Paul Fremantle, WSO2

## 1.3 Contributors (alphabetically)

Jens Jakob Andersen, NITA

Mikkel Hippe Brun, NITA

Mike Edwards, IBM

Paul Fremantle, WSO2

Thomas Gundel, IT Crew

Philip Helger, Bundesrechenzentrum

Hans Guldager Knudsen, Lenio

Bergþór Skúlason, NITA

Gert Sylvest, Avanade

## 1 Introduction

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

- 5 • No requirement to host online endpoints, hence no firewall crossing, no server infrastructure
- 6 • No requirement to support “advanced” WS-\* standards such as WS-Trust, WS-ReliableMessaging. Only  
7 minimal requirement to support WS-Security (authentication headers only).

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

## 14 1.4 Goals and non-goals

### 15 1.4.1 Goals

- 16 • Provide an interface to a message channel and relay service that supports intermittently connected  
17 systems.
- 18 • Provide access over a simple HTTPS-protected channel
- 19 • Utilize existing standards where appropriate
- 20 • Support the same message format as other BUSDOX Transport Profiles
- 21 • Lower the cost of entry for SME’s and individuals.

### 22 1.4.2 Non-Goals

- 23 • This profile does not support end-to-end security or identity. The BUSDOX Lightweight Message  
24 Exchange Profile Access Point (LIME-AP) must validate the credentials of customers using the BUSDOX  
25 LIME and map those credentials into a valid identity to be used for outbound communications.

## 26 1.5 Terminology

27 Please see Common Definitions section 2.2

## 28 1.6 Notational conventions

29 Notational conventions have been adopted from [WSDL-2.0].

30 Pseudo-schemas are provided for each component, before the description of the component. They use  
31 BNF-style conventions for attributes and elements: "?" denotes optionality (i.e. zero or one occurrences),  
32 "\*" denotes zero or more occurrences, "+" one or more occurrences, "[" and "]" are used to form groups,  
33 and "|" represents choice. Attributes are conventionally assigned a value which corresponds to their type,  
34 as defined in the normative schema. Elements with simple content are conventionally assigned a value

35 which corresponds to the type of their content, as defined in the normative schema. Pseudo schemas do  
36 not include extension points for brevity.

37

```
38 <!-- sample pseudo-schema -->
```

```
39 <defined_element
```

```
40   required_attribute_of_type_string="xs:string"
```

```
41   optional_attribute_of_type_int="xs:int"? >
```

```
42 <required_element />
```

```
43 <optional_element />?
```

```
44 <one_or_more_of_these_elements />+
```

```
45 [ <choice_1 /> | <choice_2 /> ]*
```

```
46 </defined_element>
```

### 47 **1.6.1 Normative references**

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

49 [WS-T] "Web Services Transfer (WS-Transfer)", W3C Working Draft 17 March 2009,

50 <http://www.w3.org/TR/2009/WD-ws-transfer-20090317/>

51 [WSA-1.0] "Web Services Addressing 1.0 - Core" ([http://www.w3.org/TR/2005/CR-ws-addr-core-](http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/)

52 [20050817/](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/>

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

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

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

56 [20000508/](http://www.w3.org/TR/2000/NOTE-SOAP-20000508/)

### 57 **1.6.2 Non-normative references**

58 [WSDL-2.0] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language",

59 <http://www.w3.org/TR/wsdl20/>

60

61 **1.7 Namespaces**

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

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

64

65

## 66 **2 Introduction and overview**

### 67 **2.1 Example flows**

68 The Lightweight Message Exchange Profile is designed to allow systems to participate in the BUSDOX  
69 infrastructure without needing to access service metadata or host an Access Point. Instead, they rely on an  
70 Internet Service Provider (ISP) to provide Lightweight Message Exchange Profile (LIME) services to them. A  
71 simple analogy is Internet email: Large companies may run their own Simple Mail Transport Protocol  
72 (SMTP) server and proprietary email clients to create and read messages, but individuals or small  
73 companies rely on an ISP to provide an SMTP Relay and POP3 or IMAP server.

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

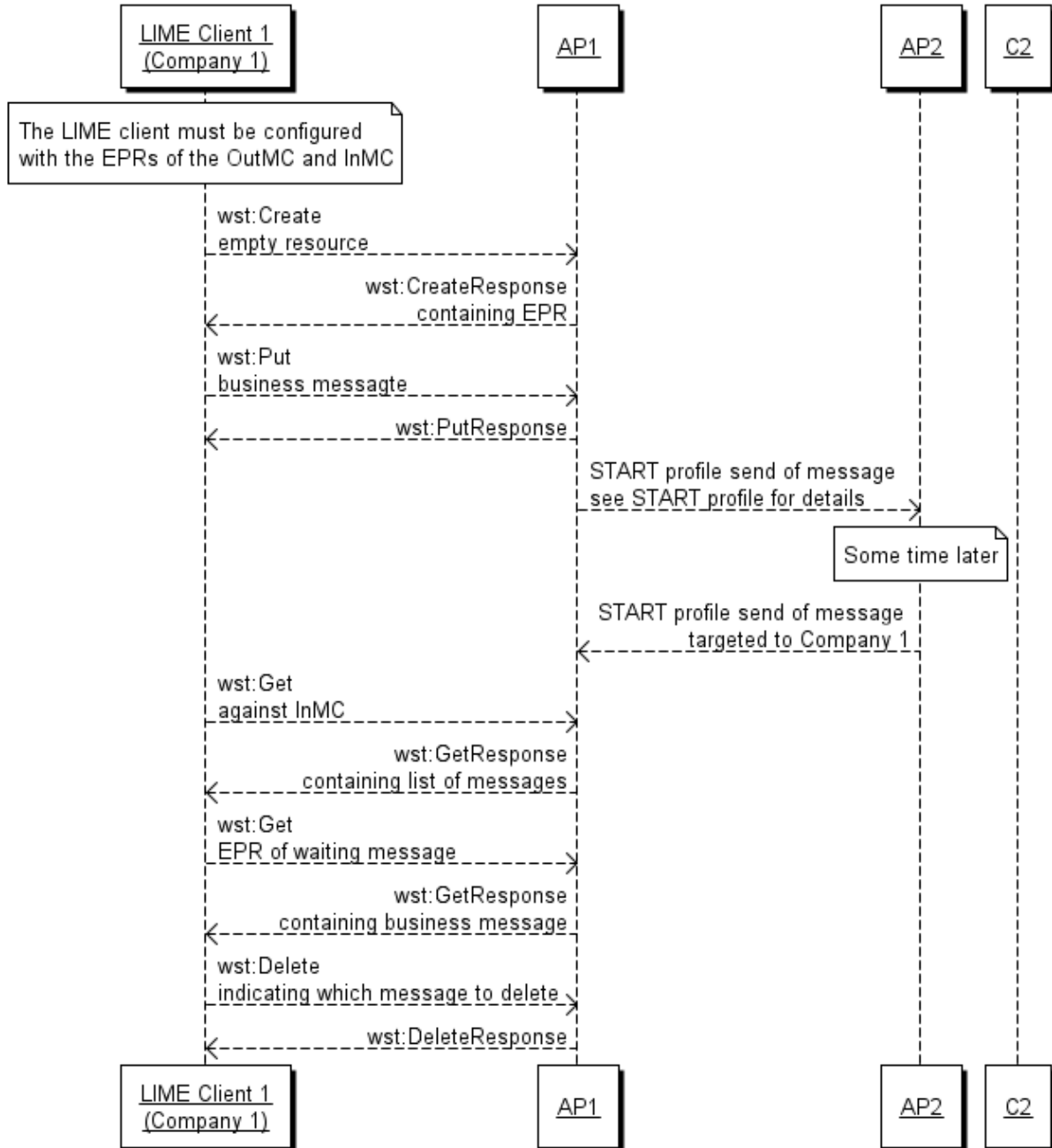
76 The diagram below shows a simple example flow. The LC needs to send a message to a company which  
77 uses an Access Point we will call AP2. However, the LC only needs to be configured to talk to a single local  
78 access point (AP). Initially the business user creates a business message using the software (out of scope for  
79 this profile). The requirements are that the business message complies with the BUSDOX specifications and  
80 that the correct business identifiers (see section 3.2) are made available to the LC.

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

85

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

88



89

90

91 **2.2 Technical Overview of the Profile**

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

- 93 • HTTPS and Basic Authentication for security
- 94 • SOAP 1.1 for the base communications
- 95 • WS-Transfer as a standard approach to accessing the message channels
- 96 • BUSDOX specific headers to define standard metadata
- 97 • BUSDOX specific XML Schema to define the message list XML format

98 Together these different technologies are used together to define a simple protocol that can allow an  
99 intermittently connected computer to fully participate in a BUSDOX infrastructure so long as they have  
100 a Lightweight Message Exchange Profile Access Point (LIME-AP) available.

101

## 102 **3 Definition of the Message Channel**

### 103 **3.1 Concepts**

104 A message channel is a WS-Transfer endpoint that either accepts or retrieves messages from an LC. A single  
105 channel may handle both incoming and outgoing messages or there may be independent channels. The  
106 profile assumes that there may be independent channels and therefore that the LC is provided with  
107 addressing information for both channels.

108 The message channel accepts outbound messages (messages from LC to AP) using the WS-Transfer Create  
109 and Put operations, and offers inbound messages (messages from AP to LC) using the WS-Transfer Get  
110 method.

### 111 **3.2 BUSDOX defined headers**

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

115 For an XML Schema for these elements, see section 5.

#### 116 **3.2.1 Recipient Business Identifier**

117 This element represents the business identifier of the ultimate recipient. This is used for service metadata  
118 lookup and message forwarding. Pseudo schema for this element is:

```
119 <ids:RecipientIdentifier scheme="xs:anyURI">xs:string</ids:RecipientIdentifier>  
120
```

121 This element contains both the Recipient identifier and identifier type.

#### 122 **3.2.2 Sender Business Identifier**

123 This element represents the business identifier of the original sender. This is used for service metadata  
124 lookup and message forwarding. Pseudo schema for this element is:

```
125 <ids:SenderIdIdentifier scheme="xs:anyURI">xs:string</ids:SenderIdIdentifier>  
126
```

127 This element contains both the Sender identifier and identifier scheme.

#### 128 **3.2.3 Document Type Identifier**

129 This element represents the type of document enclosed in the message. This is used for service metadata  
130 lookup/routing. Pseudo schema for this element is:

```
131 <ids:DocumentIdentifier scheme="xs:anyURI">xs:string</ids:DocumentIdentifier>  
132
```

133 This element contains the Document identifier and identifier scheme.

134

### 135 **3.2.4 Process Type Identifier**

136 This element represents the type of process that a document may participate in. This is used for service  
137 metadata lookup and message forwarding. Pseudo schema for this element is:

```
138 <ids:ProcessIdentifier scheme="xs:anyURI">xs:string</ids:ProcessIdentifier>
```

### 139 **3.2.5 Message Identifier**

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

```
143 <ids:MessageIdentifier>
```

144 element.

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

## 148 **3.3 Use of WS-Transfer**

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

151 *[WS-T]. Please note that we expect to update the usage of WS-Transfer to a stable WS-I compliant version*  
152 *from W3C before this specification becomes final.*

153 WS-Transfer is used to send messages from the LC to the AP as well as retrieve waiting messages.

154 The sending of messages uses the pattern we call CreatePut for reliability.

155 Receiving messages is done with two or more Gets – the first lists a page of available messages, further  
156 requests may retrieve individual messages or further pages of message listing. Messages can be DELETED  
157 once successfully retrieved. WS-Transfer does not define a resource listing model, so this profile defines a  
158 simple XML Schema for lists.

### 159 **3.3.1 Securing channels**

160 Typically the LIME-AP will secure the two message channels in the following fashion:

- 161 • The LC can list messages using the Get interface on the InMC. The LC can Get and Delete messages  
162 in the Inbound Message Channel. It cannot Create or send (Put) messages in the Inbound Message  
163 Channel.
- 164 • The LC may Create and send (Put) messages in the Outbound Message Channel. It cannot list  
165 messages, Get, or Delete messages in the OutMC.

166 In this model it is possible to have a single outbound message channel shared by many companies. This is  
167 the recommended model but it is not normative. Another alternative is that the same channel is used to  
168 both send and receive messages.

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

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

### 171 3.3.2 Use of WS-Addressing Reference Parameters

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

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

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

```
<time:ChannelIdentifier>xs:anyURI</time:ChannelIdentifier>
```

183 The MessageIdentifier is a URI which uniquely identifies a message. The message identifier is consistent  
184 across multiple hops.

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

## 186 3.4 Inbound Message Channel

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

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

```
<wsa:EndpointReference>  
  <wsa:Address>  
    http://LIME-AP.my-van.com:80/services/messagechannel  
  </wsa:Address>  
  <wsa:ReferenceParameters>  
    <ids:ChannelIdentifier>55038353</ids:ChannelIdentifier>  
  </wsa:ReferenceParameters>  
</wsa:EndpointReference>
```

194

195 The manner in which this EPR is provided to the LC is out-of-scope; it may be typed in as part of the user  
196 configuration.

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

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

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

206 Each individual message in a channel has an Endpoint Reference which contains both the ChannelIdentifier  
207 as well as a unique MessageIdentifier as reference parameters.

208 Here is an example of an Endpoint Reference for a message.

```
209 <wsa:EndpointReference>  
210   <wsa:Address>  
211     http://LIME-AP.van.co.uk:80/services/transfer  
212   </wsa:Address>  
213   <wsa:ReferenceParameters>  
214     <ids:ChannelIdentifier>55038353</ids:ChannelIdentifier>  
215     <ids:MessageIdentifier>uuid:f8290-4321kj2349-  
216     8aiuyfga0</ids:MessageIdentifier>  
217   </wsa:ReferenceParameters>  
218 </wsa:EndpointReference>
```

### 219 3.4.1 Finding available messages

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

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

227 The LC may adopt two strategies for downloading messages:

- 228 • The LC may simply retrieve the first page, download all listed messages, delete them, and then  
229 relist. The next page will then be listed.
- 230 • The LC may retrieve the first page of listed messages. At the end of this list is the EPR of the next  
231 page. The LC may then retrieve the next page, and so forth until it has listed all available messages.

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

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

236 **/lime:PageList**

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

239 **/ lime:PageList /lime:EntryList**

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

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

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

246 **/lime:PageList/lime:EntryList /lime:Entry**

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

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

249 This required attribute captures the size of the document waiting for collection, in Kilobytes  
250 (multiples of 1024), as a numeric value.

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

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

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

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

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

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

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

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

267

268 Example:

```
<ids:ChannelIdentifier wsa:isReferenceParameter="true">Channel  
Identifier</ids:ChannelIdentifier>  
<lime:PageIdentifier wsa:isReferenceParameter="true">  
5  
</lime:PageIdentifier>
```

269

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

271 For the purposes of this profile, the EndpointReferences returned in the sequence of Entries MUST contain

272 the following two reference parameters:

```
<lime:ChannelIdentifier>xs:String</lime:ChannelIdentifier>  
<ids:MessageIdentifier>xs:String</ids:MessageIdentifier>
```

273

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

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
  An sample XML response to the page listing GET request
-->
<lime:PageList
  xmlns:lime="http://busdox.org/transport/lime/1.0/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
  numberOfEntries="1">
  <lime:EntryList>
    <lime:Entry size="8295" creationTime="2009-02-18T12:33:45Z"
      messageBodyLocalName="Order"
      messageBodyNamespace="http://busdox.org/ns/Order">
      <wsa:EndpointReference>
        <wsa:Address>
          http://LIME-AP.my-van.com:80/services/transfer
        </wsa:Address>
        <wsa:ReferenceParameters>
          <ids:ChannelIdentifier>55082098</ids:ChannelIdentifier>
          <ids:MessageIdentifier>uuid:45989-2429-
132412312</ids:MessageIdentifier>
        </wsa:ReferenceParameters>
      </wsa:EndpointReference>
    </lime:Entry>

  </lime:EntryList>

  <lime:NextPageIdentifier>
    <wsa:EndpointReference>
      <wsa:Address>
        http://LIME-AP.my-van.com:80/services/messagechannel
      </wsa:Address>
      <wsa:ReferenceParameters>
        <ids:ChannelIdentifier>55038353</ids:ChannelIdentifier>
      <!--
        NOTE: The 'PageIdentifier' may be replaced by element in
        any namespace that represents a system-specific ID of the next
page
      -->
      <PageIdentifier
xmlns="http://someNamespace.org">2</PageIdentifier>
    </wsa:ReferenceParameters>
  </wsa:EndpointReference>
</lime:NextPageIdentifier>

</lime:PageList>

```

275

276

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

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

#### 281 **3.4.2 Getting a message using WS-Transfer**

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

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

#### 286 **3.4.3 Inclusion of SAML attributes**

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

291 The following header contains the SAML attribute:

```
292 <lime:identityAssurance>
```

```
293   <saml2:Attribute/>
```

```
294 </lime:identityAssurance>
```

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

#### 297 **3.4.4 Deleting messages**

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

### 301 **3.5 The Outbound Message Channel**

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

### 305 **3.6 Message Sending**

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

308

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

311 1. In the first stage, the LC creates an *empty* resource in the OutMC, not containing the real  
312 message. This is done using the CREATE request and with no business message

313 <wst:Create/>

314 The response to this is:

315 <wxf:ResourceCreated>endpoint-reference</wxf:ResourceCreated>

316 which contains an EndpointReference of the resource that will be used to transmit the message.

317 The EndpointReference will contain a unique ids:MessageIdentifier reference parameter.

318 If this operation is incomplete or the response is dropped at the network level, the LC should

319 retry. In this case, there may be an extra unused Resource and EPR available on the LIME-AP

320 (from the first failed CREATE request). The LIME-AP SHOULD keep such resources available for up

321 to one hour, to allow for timing issues in the LC. The LIME-AP SHOULD garbage collect/delete any

322 such resources that remain empty for extended periods of time.

323

324 The Create message MUST include the following BUSDOX defined headers in the SOAP Header:

325 *ids:RecipientIdentifier, ids:ChannelIdentifier, ids:SenderIdentifier, ids:DocumentIdentifier,*

326 *ids:ProcessIdentifier*

327

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

329

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

331

332 The client MAY include other headers in the message. Any headers that are not defined in the

333 BUSDOX-namespace and are not part of the Reference Parameters SHOULD be stored and

334 SHOULD be relayed onward with the business message.

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

337

338 The Put message request SHOULD NOT include the BUSDOX defined headers in the SOAP Header,

339 except the reference parameters. Other non-BUSDOX headers SHOULD NOT be included. Any

340 headers that are not defined in the BUSDOX-namespace and are not part of the Reference

341 Parameters SHOULD be dropped by the Access Point before transferring to another Access Point.

342

343 The LC SHOULD repeat this second step as often as required until it gets a successful response. If

344 there is a long period of time between step 1 and step 2, it is possible that the LIME-AP has

345 deleted the resource. In this case the LIME-AP will return a fault to the LC indicating an unknown

346 EPR. In this case, the LC SHOULD restart at step 1.

347

348 The LC SHOULD log the PUT request message and PUT response message for proof-of-delivery.

349 The LIME-AP MUST include the WS-Addressing RelatesTo header.

350

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

```
<wsa:EndpointReference>  
  <wsa:Address>  
    http://LIME-AP.my-van.com:80/services/messagechannel  
  </wsa:Address>  
  <wsa:ReferenceParameters>  
    <ids:ChannelIdentifier>outbound</ids:ChannelIdentifier>  
  </wsa:ReferenceParameters>  
</wsa:EndpointReference>
```

354

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

```
<wxf:ResourceCreated>  
  <wsa:EndpointReference>  
    <wsa:Address>  
      http://LIME-AP.my-van.com:80/services/messagechannel  
    </wsa:Address>  
    <wsa:ReferenceParameters>  
      <lime:ChannelIdentifier>outbound</lime:ChannelIdentifier>  
      <ids:MessageIdentifier>uuid:45989-2429-  
357 132412313</ids:MessageIdentifier>  
358 132412313</ids:MessageIdentifier>  
359 132412313</ids:MessageIdentifier>  
360 132412313</ids:MessageIdentifier>  
361 132412313</ids:MessageIdentifier>  
362 132412313</ids:MessageIdentifier>  
363 132412313</ids:MessageIdentifier>  
364 132412313</ids:MessageIdentifier>  
365 132412313</ids:MessageIdentifier>  
366 132412313</ids:MessageIdentifier>  
367 132412313</ids:MessageIdentifier>  
368 132412313</ids:MessageIdentifier>  
</wxf:ResourceCreated>
```

369

370  
371 Once the LIME-AP receives a complete successful Put operation it can relay the message on to the final  
372 recipient. The LIME-AP should treat subsequent Puts of the same message as correct, as these indicate that  
373 the LC has not yet received a successful response and will keep retrying. The LIME-AP SHOULD use the  
374 <ids:MessageIdentifier> of the Put request when relaying messages onward.

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

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

### 378 3.6.1 Faults

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

383

384 **Channel Full Fault**

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

385

386 **Unknown Endpoint**

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

387 **Security Error**

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

388

389 **Server Error**

[action]	<a href="http://busdox.org/2010/02/channel/fault">http://busdox.org/2010/02/channel/fault</a>
Code	s:Sender
Subcode	lime:ServerError
Reason	ServerError

Detail	As detailed by the AP
--------	-----------------------

390  
391

### 392 **3.7 Use of HTTP**

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

### 394 **3.8 Use of MTOM**

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

## 399 **4 Security**

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

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

407 The minimum required security for LIME is to use:

- 408 • HTTP Basic authentication
- 409 • Transport Layer Security for encryption

410 The WS-SecurityPolicy for this is as follows:

```
411 <sp:HttpsToken >  
412 <wsp:Policy >  
413   <sp:HttpBasicAuthentication />  
414 </wsp:Policy>  
415 </sp:HttpsToken>
```

416

417 **5 Appendix B – XML Schema for headers**

418 For description of and schema for the identifiers used in the headers, see [BDEN-CDEF].

419

420 **6 Appendix C – XML Schema for PageList XML**

421 XSD for the PageList XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://busdox.org/transport/lime/1.0/"
  elementFormDefault="qualified"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://busdox.org/transport/lime/1.0/"
  xmlns:wsa="http://www.w3.org/2005/08/addressing" version="0.9.5">

  <import namespace="http://www.w3.org/2005/08/addressing"
  schemaLocation="ws-addr.xsd" />

  <element name="PageList" type="tns:PageListType" />

  <complexType name="PageListType">
    <sequence>
      <element name="EntryList" type="tns:EntryListType" />
      <element name="NextPageIdentifier"
type="tns:NextPageIdentifierType" minOccurs="0" />
    </sequence>
    <attribute name="numberOfEntries" type="long"
use="optional"></attribute>
  </complexType>

  <complexType name="EntryListType">
    <sequence>
      <element name="Entry" type="tns:Entry" minOccurs="0"
maxOccurs="unbounded" />
    </sequence>
  </complexType>

  <complexType name="Entry">
    <sequence>
      <element ref="wsa:EndpointReference" />
    </sequence>
    <attribute name="size" type="long"></attribute>
    <attribute name="creationTime" type="dateTime"
use="optional"></attribute>
    <attribute name="messageBodyLocalName" type="string"
use="optional"></attribute>
    <attribute name="messageBodyNamespace" type="anyURI"
use="optional"></attribute>
  </complexType>

  <complexType name="NextPageIdentifierType">
    <sequence>
      <element ref="wsa:EndpointReference"/>
    </sequence>
  </complexType>
</schema>
```