



# Business Document Exchange Network - Common Definitions

---

*Version 1.0.0*



WP8 2009-11-27



## 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
2.2	Terminology .....	5
2.2.1	Indication of Requirement Levels .....	5
2.2.2	Common terms .....	5
2.2.3	Notational conventions.....	7
2.2.4	Normative references .....	7
2.2.5	Non-normative references.....	7
2.3	Namespaces.....	8
3	Identifiers.....	9
3.1	Notational conventions.....	9
3.2	On the use of percent encoding in URLs.....	9
3.3	On Scheme Identifiers.....	9
3.4	Participant Identifier.....	10
3.4.1	Universal Participant Identifier Scheme .....	10
3.4.2	XML format for Participant Identifiers.....	10
3.4.3	Using participant identifiers in URLs.....	11
3.5	DocumentIdentifier.....	11
3.5.1	XML Representation of Document Identifiers .....	12
3.5.2	URL representation of Document Identifiers.....	13
3.6	Process Identifiers.....	14
3.6.1	XML Representation of Process Identifiers.....	14
3.7	BUSDOX defined identifiers .....	15
3.7.1	Recipient Participant Identifier.....	16
3.7.2	Sender Participant Identifier.....	16

3.7.3	Document Type Identifier .....	16
3.7.4	Process Type Identifier.....	16
3.7.5	Message Identifier .....	16
3.7.6	Channel Identifier .....	16
3.8	Basic Profile.....	17
3.9	SOAP 1.1.....	17
3.10	Use of HTTP.....	17
3.11	WS-Addressing 1.0.....	17
3.11.1	The <wsa:To> Header .....	17
3.11.2	The <wsa:MessageIdentifier> Header .....	17
3.11.3	The <wsa:RelatesTo> Header.....	17
3.11.4	EndpointReferences.....	17
4	Appendix A.....	20
5	.....	20
5.1	XML Schema for message identifiers.....	20

# 1 Document information

## 1.1 Document history

Date	Version	Initials	Changes
2009-06-04	0.81	GS	First version
2009-06-25	0.82	GS	Updated terminology
2009-08-31	0.9.0	GS	Updated reference list, terminology
2009-10-21	0.9.5	GS	Changed "OIOUBL-2.02" to "UBL-2.0", and profile identifiers from current URL format to CEN/BII profile names, e.g. "BII03", changed the name of the 'no process' identifier to "busdox:noprocess", updated schema for identifiers.
2009-11-17	0.9.6	GS	Changed identifier formats. Changed <a href="http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifier/1.0/">http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifier/1.0/</a> to "busdox_actorid_upis", " <a href="http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/">http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/</a> " to "busdox_docid_qns", and " <a href="http://busdox.org/processIdentifiers/1.0/UBL/CENBII/1.0/">http://busdox.org/processIdentifiers/1.0/UBL/CENBII/1.0/</a> " to "cenbii_procid_ubl".
2009-11-24	0.9.7	GS	Changed 'Party' to 'Participant' everywhere, updated schema & all examples. Updated identifiers from the NNN_NNN_NNN format to NNN-NNN-NNN
2009-11-27	1.0.0	GS	Final updates

## 1.2 Editor

Gert Sylvest, Avanade

## 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

## 1 **2 Introduction**

2 This document contains the definitions and terms that are common between the Business Document  
3 Exchange Network (BUSDOX) service metadata and transport specifications. These are:

- 4 • The START and LIME transport specifications
- 5 • The SML (Service Metadata Locator) and SMP (Service Metadata Publishing) specifications

### 6 **2.1 Goals and non-goals**

7 The goal of this document is to describe the following:

- 8 • Define the extensible scheme format of Participant Identifiers
- 9 • Define the extensible scheme format of Document Type Identifiers
- 10 • Define the scheme format of Profile/Process Identifiers
- 11 • Define terms that are common among specifications, such as 'Access Point'
- 12 • Describe the notational conventions that apply to all Transport Infrastructure Specifications
- 13 • Define the common Schemas for headers in the START (Secure Trusted Asynchronous Reliable  
14 Transport) and LIME (Lightweight Message Exchange Profile) transport profiles
- 15 • Describe the use of WS-Addressing headers common to all SOAP based BUSDOX transport profiles
- 16 • Describe the relation to WS-I basic profile 1.1 requirements

### 18 **2.2 Terminology**

19 Terminology use applies to all BUSDOX Transport Infrastructure Specifications.

#### 20 **2.2.1 Indication of Requirement Levels**

21 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT",  
22 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

#### 23 **2.2.2 Common terms**

24 *Business Document Exchange Network (BUSDOX) Infrastructure Sphere*: the set of peer Access Points and  
25 Service Metadata Publishing and Locator services that:

- 26 1. Meet the governance requirements of any BUSDOX infrastructure instance
- 27 2. Are listed as endpoints by BUSDOX Service Metadata Publishers
- 28 3. Are accessible under service level agreements for APs defined in any instance of a BUSDOX  
29 infrastructure

30 *BUSDOX Access Point/AP*: a peer in a BUSDOX Infrastructure instance.

31 *SAML Token/Assertion*: The terms SAML token and SAML assertion will be used interchangeably.

32 *Source Access Point/SrcAP*: An Access Point sending a message to another Access Point

33 *Destination Access Point/DestAP*: An Access Point receiving a message from a SrcAP.

34 *Secure Trusted Asynchronous Reliable Transport/START*: The Secure Trusted Asynchronous Reliable  
35 Transport, see the START transport specification.

36 *Lightweight Message Exchange/LIME*: The Lightweight Message Exchange Transport, see the LIME  
37 transport specification.

38 *Lightweight Client/LC*: Client application that communicates with a LIME compliant Access Point (LIME-AP)

39 *Lightweight Profile Access Point/LIME-AP*: A BUSDOX Access Point that exposes the Lightweight Profile  
40 Interface towards client applications. An LIME-AP may be an existing VAN or a new service offered by  
41 governments or private companies.

42 *Message Channel/MC*: An LIME-AP offers a Message Channel (**MC**) interface to the LC. There are (at least)  
43 two MCs available to the LC.

44 *Inbound/Outbound Message Channel/InMC/OutMC*: An Inbound Message Channel stores messages  
45 destined for the LC, and the Outbound Message Channel is used by the LC as a relay for messages destined  
46 for other companies.

47 *Endpoint Reference/EPR*: Each Message Channel of a LIME-AP is uniquely identified by a WS-Addressing  
48 Endpoint Reference (EPR)

49 *Channel Identifier/ChannelID*: A channel of an LIME-AP.

50 *Service Metadata Locator service/SML*: A service which provides a client with the capability of discovering  
51 the Service Metadata Publisher endpoint associated with a particular participant identifier. A client uses  
52 this service in order to find where information is held about services for a particular participant business.

53 *Service Metadata Publisher/SMP*: A service metadata publisher offers a service on the network where  
54 information about services of specific participant businesses can be found and retrieved. It is necessary for  
55 a client application to retrieve the metadata about the services for a target participant business before the  
56 client can use those services to send messages to the participant business.

57 *Service Metadata Consumer/SMC*: A Service Metadata Consumer is any entity consuming Service Metadata  
58 provided by a Service Metadata Publisher. This is typically the sender of the document, or the sender side  
59 Access Point.

60 *Participant Identifier*: A participant business level identifier such as GLN (a GS1 Global Location Number) or  
61 DUNS (Dun & Bradstreet) number that is used to identify a trading partner. In the context of BUSDOX,  
62 participant identifiers are used to discover services associated with trading partners in Service Metadata.

63 *Secure Token Service/STS*: a service that offers security tokens to network clients.

64 *Transport Layer Security/TLS*: Transport Layer Security is the standard most used to secure HTTP. It is the  
65 successor to Secure Sockets Layer (SSL).

66 *Service Metadata Publisher Certificate (SMP Certificate)*: A certificate used by a specific SMP to create all  
67 signatures of the signed resources.

### 68 2.2.3 Notational conventions

69 Notational conventions have been adopted from **Error! Reference source not found.** and apply to all  
70 BUSDOX Transport Infrastructure Specifications.

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

```
78  
79 <!-- sample pseudo-schema -->  
80 <defined_element  
81     required_attribute_of_type_string="xs:string"  
82     optional_attribute_of_type_int="xs:int"? >  
83 <required_element />  
84 <optional_element />?  
85 <one_or_more_of_these_elements />+  
86 [ <choice_1 /> | <choice_2 /> ]*  
87 </defined_element>
```

### 88 2.2.4 Normative references

89 This section only covers references to documents referred to from this document.

90 [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/>

92 [SOAP-1.1] "SOAP Version 1.1", <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

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

95 [BP-1.1] "Basic Profile Version 1.1", <http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>

### 96 2.2.5 Non-normative references

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

99 [RFC4122] "A Universally Unique Identifier (UUID) URN Namespace", <http://www.ietf.org/rfc/rfc4122.txt>

100

101

102 **2.3 Namespaces**

103 The following table lists XML namespaces that are used in this document. The choice of any namespace  
104 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>
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>

105

106

## 107 **3 Identifiers**

108 This section defines what participant business-, document- and process-identifiers are, and how they are  
109 represented in XML elements and URLs.

### 110 **3.1 Notational conventions**

111 For describing the textual format of identifiers, the following conventions are used:

112 *Everything within the curly brackets {} can be substituted by specific values.*

113 *Everything with square brackets [] represents optional content, whether literals or not.*

114 *Everything outside the curly brackets must be treated as literals.*

115 For example, for an identifier with the value “0010:5798000000001”, the format definition

116 `/{identifier}/service[/endpointName]`

117 Can be instantiated to either of the strings

118 `/0010:5798000000001/service`

119 And

120 `/0010:5798000000001/service/endpointName`

### 121 **3.2 On the use of percent encoding in URLs**

122 When any types of BUSDOX identifiers are used in URLs, each section between slashes MUST be percent  
123 encoded individually, i.e. section by section, according to **Error! Reference source not found.**

124 For example, this implies that for an URL in the form of “/{identifier scheme}::{id}/services/{docType}”, the  
125 slash literals MUST NOT be URL encoded.

### 126 **3.3 On Scheme Identifiers**

127 Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be  
128 defined outside of the BUSDOX specifications. BUSDOX defines a few generally usable identifiers, but any  
129 instance of the BUSDOX infrastructure may choose to define identifier schemes that match the type of  
130 documents, participants or profiles that are relevant to support in that instance.

131 Any Scheme Identifier defined outside of this specification MUST take the form <domain>-<identifierArea>-  
132 <identifier type>, such as for example “busdox-actorid-upis”.

- 133 • Domain: BUSDOX
- 134 • Identifier area: Actor Identifiers (“actorid”)
- 135 • Identifier type: “Universal Participant Identifier Scheme” (upis)

136 Scheme identifiers are used in the SML spec for creating DNS aliases for SMPs, and may therefore only  
137 contain the following characters:

138 [a-z], [A-Z], [0-9], [-]

139 A scheme identifier SHOULD be as short as possible, and MUST NOT exceed 25 characters.

### 140 **3.4 Participant Identifier**

141 Participant identifiers logically consist of a *scheme identifier* and the *participant identifier* itself. Participant  
142 identifiers are associated with groups of services, or Service Metadata.

143 The *scheme identifier* indicates the specification of the participant identifier format, i.e. its representation  
144 and meaning. Currently there is only one scheme defined, the *Universal Participant Identifier Scheme*, and  
145 is identified by the string literal "busdox-actorid-upis".

#### 146 **3.4.1 Universal Participant Identifier Scheme**

147 This specification only describes the 'Universal Participant Identifier scheme', although other domain-  
148 specific schemes may be defined.

149 The scheme indicates the format follow identifier format:

150 {type identifier}:{participant identifier}.

151 The type identifier is 4-digit number indicating the type of participant identifier, such as GLN, DUNS, CVR or  
152 another scheme. This scheme is dynamically extendable and will be decided outside of this specification.  
153 Note that the {type identifier} part of the participant identifier *is not* equivalent to the scheme of the  
154 identifier - the {type identifier} indicates the *type of participant identifier*, whereas the scheme identifier  
155 indicates the format and semantics of an identifier string.

156 For the moment, only the type identifier "0010" has been defined as indicating a GLN number, such as for  
157 example "0010:5798000000001".

#### 158 **3.4.2 XML format for Participant Identifiers**

159 The <ParticipantIdentifier> element is used to represent participant identifiers and scheme information.

160 Pseudo-scheme for ParticipantIdentifier:

```
161 <smp:ParticipantIdentifier scheme="xs:string">  
162   xs:string  
163 </smp:ParticipantIdentifier>
```

164

165 Where the 'scheme' attribute indicates the scheme of the participant identifier.

Field	Description
ParticipantIdentifier	A participant identifier which may be associated with a group of services.
ParticipantIdentifier/ @scheme	The scheme of the participant identifier. This scheme indicates the <i>format</i> of the participant identifier (i.e. the textual format) – not its semantic type (e.g. DUNS or GLN).  This scheme type MUST be in the form of a URI.  The URI that indicates the “Universal Participant Identifier” format is “busdox-actorid-upis”  When processing a participant identifier in XML format, it MUST be treated as case insensitive.

166

### 167 3.4.3 Using participant identifiers in URLs

168 The following format is used:

169 {identifier scheme}::{id}

170 Where 'identifier scheme' is the format of the identifier, and 'id' is the participant identifier itself, following  
171 the format indicated by the 'identifier type' part. The {id} could for example be the format defined as  
172 'Universal Participant Identifier' scheme.

173 In a URL, the string represented by “{identifier scheme}::{id}” MUST be percent encoded following **Error!**  
174 **Reference source not found.**, and the guidelines given above.

175 Non-normative example that uses the Universal Participant Identifier Format, assuming the participant  
176 identifier “0010:5798000000001”:

177 *busdox-actorid-upis::0010:5798000000001*

178 In percent encoded form:

179 *busdox-actorid-upis%3a%3a0010%3a5798000000001*

180 When processing a participant identifier in an URL, it MUST be treated as case insensitive. Note that any  
181 surrounding slashes which belong to the URL rather than the various identifiers (which may take the forms  
182 of URLs) are *not* percent encoded.

### 183 3.5 DocumentIdentifier

184 Documents are represented by an identifier (identifying the document type) and a scheme type which  
185 represents the scheme or format of the identifier itself. It is outside the scope of this document to list  
186 identifier schemes that may be valid in the BUSDOX context.

187 This specification defines a single identifier scheme, the 'QName/Subtype Identifier Scheme', which is  
188 identified by the following URI:

189 *busdox-docid-qns*

190 This scheme is based on a concatenation of the document namespace, root element, and optional (and  
191 document-dependent) subtype:

192 {rootNamespace}::{documentElementLocalName}[##{Subtype identifier}]

193 Where '[' ]' denotes an optional part of the identifier, and everything outside '{ }' are string literals.

194 For example, in the case of a NES-UBL order, this document can then be identified by

- 195 • **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2
- 196 • **Document element local name:** Order
- 197 • **Subtype identifier:** UBL-2.0 (since several versions of the Order schema may use the same  
198 namespace + document element name)

199 The document type identifier will then be:

200 urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0

### 201 3.5.1 XML Representation of Document Identifiers

202 The <DocumentIdentifier> element is used to represent document identifiers and scheme information.

203 Pseudo-scheme for DocumentIdentifier:

204 <DocumentIdentifier scheme="xs:string">xs:string</DocumentIdentifier>  
205

206 Where the 'scheme' attribute indicates the scheme of the document identifier.

Field	Description
DocumentIdentifier	A document identifier representing a specific document type. In the case of a NES-UBL order document, this would be "urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0".
ParticipantIdentifier/@scheme	The scheme of the document identifier. This scheme identifier MUST be in the form of a URI. This document defines the 'QName/Subtype Identifier Scheme', which is identified by the following URI:  <i>busdox-docid-qns</i>  When processing a document identifier in XML format, it MUST be treated as case insensitive.

207

208 **3.5.2 URL representation of Document Identifiers**

209 When representing document identifiers in URLs, the document identifier itself will be prefixed with the  
210 scheme identifier. For example, the 'QName/Subtype Identifier Scheme' is indicated by this identifier:

211 "busdox-docid-qns".

212 The format of this is:

213 {identifier scheme}::{id}

214 In the case that the 'QName/Subtype Identifier Scheme' is used, the complete format is:

215 {identifier scheme}::{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}].

216 As a non-normative example, in the case of a NES-UBL order, this document can then be identified by

- 217
- 218 • **Identifier scheme:** busdox-docid-qns
  - 219 • **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2
  - 220 • **Document element local name:** Order
  - 221 • **Subtype identifier:** UBL-2.0 (since several versions of the Order schema may use the same  
namespace + document element name)

222 The document type identifier will then be:

223 busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0

224 Rules for parsing this identifier:

- 225
- 226 • The text up until the first "::" is the identifier scheme identifier
  - 227 • The text before the next to last "::" and last "::" is the root namespace
  - 228 • The text between the last occurrence of ':' and last occurrence of '##' OR end of the string is the  
document element local name.
  - 229 • The text following the first '##' after the document element local name (if any) is the subtype  
230 identifier.

231 Note that although namespaces and element names are case sensitive, the document identifier MUST be  
232 treated as case insensitive.

233 This string must be percent encoded (see **Error! Reference source not found.**, sect. 2.1) if used in an URL. In  
234 that case, the above identifier will then read as:

235 *busdox-docid-qns%3a%3aurn%3aoasis%3anames%3aspecification%3aubl%3aschema%3axsd%3aOrder-*  
236 *2%3a%3aOrder%23%23UBL-2.0*

237 Note the limitation that XML document types with the following characteristics MUST NOT be referenced  
238 using Service Metadata Publishing:

- 239 • Documents with only local names (i.e. without namespaces)
- 240 • Documents that need to be identified with a subtype identifier, and where the subtype part of the  
241 identifier does not correspond to a specific, mandatory attribute value or element value in the  
242 document that is based on XML Schema simple content.

## 243 **3.6 Process Identifiers**

244 A process identifier represents a process that a specific document type can participate in. Process  
245 identifiers consist of the process identifier itself, and a scheme or identifier format type. As for the other  
246 schemed identifier types, additional process identifier schemes may be defined outside of the BUSDOX  
247 specifications.

### 248 **3.6.1 XML Representation of Process Identifiers**

249 Pseudo-schema for the ProcessIdentifier XML element:

```
250 <ProcessIdentifier scheme="xs:string">xs:string</ProcessIdentifier>
```

251

252

253 Description of the individual fields (elements and attributes).  
 254

Field	Description
ProcessIdentifier	<p>The identifier of the process. A process is identified by a string that is defined outside of this specification. For example, the CEN BII may choose to indicate a UBL-based ‘simple procurement’ process (or ‘profile’ in UBL terminology) with the identifier “BII07”, or UBL-based basic invoice exchange profile with the identifier “BII04”.</p> <p>This document just defines one process identifier, which represents documents that are <i>not</i> sent under any specific process:</p> <p style="text-align: center;">busdox:noprocess</p> <p>The process identifier <b>MUST</b> be treated as case insensitive. This identifier <b>MUST</b> be used with the process scheme “busdox-procid-transport”</p>
ProcessIdentifier/@scheme	<p>Indicates the format of the process identifier. The format of this identifier may be different from domain to domain – for example, in a domain where BPEL definitions of processes exist, a technical identifier derived from the BPEL definitions may be used, whereas a taxonomy may be created in another domain. Processes (or profiles) defined by the CEN BII workshop could for example choose to use the identifier “cenbii-procid-ubl” to indicate the format of process identifiers.</p> <p>This document just defines one process scheme identifier, which represents transport-specific process identifiers:</p> <p style="text-align: center;">busdox-procid-transport</p> <p>Currently the only valid process identifier under this scheme is the identifier “busdox:noprocess”, which indicates that a message is not sent under any named process.</p>

255  
 256

257 **3.7 BUSDOX defined identifiers**

258 The identifiers defined in this section are used both with the Service Metadata specifications and the  
 259 transport specifications. Every BUSDOX message has associated metadata included, in the form of headers,  
 260 so that Access Points can route messages without relying on knowledge of the message payload.

261 For an XML Schema for these elements, see the section “**Error! Reference source not found.**”.

262 **3.7.1 Recipient Participant Identifier**

263 This element represents the participant identifier of the ultimate recipient. This is used for Service  
264 Metadata Lookup and message forwarding. Pseudo schema for this element is:

265 `<ids:RecipientIdentifier scheme="xs:string">xs:string</ids:RecipientIdentifier>`

266 This element contains both the Recipient identifier and identifier scheme. It follows the rules for XML  
267 representation of participant identifiers laid out in section **Error! Reference source not found..**

268 **3.7.2 Sender Participant Identifier**

269 This element represents the participant identifier of the original sender. This is used for Service Metadata  
270 lookup and message forwarding. Pseudo schema for this element is:

271 `<ids:SenderIdIdentifier scheme="xs:string">xs:string</ids:SenderIdIdentifier>`

272 This element contains both the Sender identifier and identifier scheme. This element contains both the  
273 Recipient identifier and identifier type. It follows the rules for XML representation of participant identifiers  
274 laid out in section **Error! Reference source not found..**

275 **3.7.3 Document Type Identifier**

276 This element represents the type of document enclosed in the message. This is used for Service Metadata  
277 lookup/routing. Pseudo schema for this element is:

278 `<ids:DocumentIdentifier scheme="xs:string">xs:string</ids:DocumentIdentifier>`

279 This element contains the Document identifier and identifier scheme. It follows the rules for XML  
280 representation of document identifiers laid out in section **Error! Reference source not found..**

281 **3.7.4 Process Type Identifier**

282 This element represents the type of process that a document may participate in. This is used for Service  
283 Metadata lookup. Pseudo schema for this element is:

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

285 This element follows the rules for XML representation of process identifiers laid out in section 3.6.

286 **3.7.5 Message Identifier**

287 Because BUSDOX Messages may pass between several parties (for example in the "four-corner" model,  
288 from LIME client1 to AP1 to AP2 to LIME client2), it is desirable to have a constant message identifier that  
289 uniquely identifies the message across multiple hops, for tracing purposes. This message identifier is  
290 contained in the 'MessageIdentifier' element:

291 `<ids:MessageIdentifier>xs:string</ids:MessageIdentifier>`

292 **3.7.6 Channel Identifier**

293 Channel identifiers are used with the START and LIME profiles in order to discern between multiple  
294 channels of communication within each AP. It has the following form:

295 `<ids:ChannelIdentifier>xs:string</ids:ChannelIdentifier>`

296 **3.8 Basic Profile**

297 The conformance Requirements in the WS-I Basic Profile 1.1 specification MUST be followed.

298 **3.9 SOAP 1.1**

299 APs MUST use SOAP 1.1 for all message exchange.

300 Messages MUST use the document/literal style.

301 **3.10 Use of HTTP**

302 Messages MUST use the “text/xml” media type, unless MTOM is being used. If MTOM is in place, the HTTP  
303 media type MUST be “multipart/related”, and the content-type of the part containing the SOAP message  
304 MUST be “text/xml”.

305 The SOAPAction HTTP Header MUST be used, and MUST correspond to the WS-Addressing ‘Action’  
306 attribute.

307 **3.11 WS-Addressing 1.0**

308 This profile only supports the use of WS-Addressing 1.0.

309 **3.11.1 The <wsa:To> Header**

310 All outgoing message interactions MUST include a wsa:To header. Responses MAY omit this header.

311 **3.11.2 The <wsa:MessageIdentifier> Header**

312 All message interactions MUST include a wsa:MessageIdentifier header.

313 Any participant that assigns a value to a <wsa:MessageIdentifier> header block MUST ensure that there is  
314 negligible probability that that participant or any other participant will accidentally assign the same  
315 identifier to any other message. Parties SHOULD use the UUID standard (see [RFC4122] )for generating  
316 message Ids.

317 Note that [WSA-1.0] requires that <wsa:MessageIdentifier> values be absolute IRIs.

318 **3.11.3 The <wsa:RelatesTo> Header**

319 Responses MUST have exactly one <wsa:RelatesTo> header present, and if the RelationshipType attribute is  
320 available, it MUST be set to the value “http://www.w3.org/2005/08/addressing/reply”.

321 **3.11.4 EndpointReferences**

322 EndpointReferences MUST NOT have <wsa:Metadata> elements.

323 **4 Undeliverable Messages**

324 There may be situations where an Access Point is unable to deliver the message sent by either the LIME  
325 client or another START Access Point. These include but are not limited to:

- 326
- The Recipient ID is not listed in the SML
- 327
- The particular document type or process is not listed in the SMP

- 328 • The recipient AP is not contactable over a long period of time.
- 329 • The recipient AP is contactable but the message cannot be delivered due to faults or WSRM  
330 Sequence problems.
- 331 • The certificate of the receiving AP has been revoked.
- 332 • A queue at the Access Point is full.

333 In these cases, the AP at which the error is identified SHOULD create a business message targeted at the  
334 sender identifier which conveys that the message is undeliverable. An example of this message is:

```
335 <ids:MessageUndeliverable>
336   <ids:MessageIdentifier>
337     uuid:45989-2429-132412313
338   </ids:MessageIdentifier>
339   <ids:ReasonCode>METADATA_ERROR</ids:ReasonCode>
340   <ids:Details>Some further details here</ids:Details>
341 </ids:MessageUndeliverable>
342
```

343 When sending this message, the AP MUST use the following process:

- 344 1. Look up the sender identifier together with the proscribed document identifier and process identifier  
345 (see below) in the Service Metadata Locator and Service Metadata Publisher according to the normal  
346 BUSDOX approach.
- 347 2. If there is no entry for the sender identifier or the MessageUndeliverable document, then no message  
348 will be sent and the sender will not be notified of the MessageUndeliverable status. In this case, the AP  
349 SHOULD log this locally for the benefit of system administrators.
- 350 3. If the AP has an identifier in the BUSDOX network, then the AP MUST use this identifier for sending the  
351 MessageUndeliverable message. However, it is not a requirement that APs have BUSDOX participant  
352 identifiers. In the case that the AP does not have a participant identifier, then the message should have  
353 the following fixed value for the ids:SenderId header:

```
354 <smp:ParticipantIdentifier scheme="busdox-actorid-transport">
355   busdox:sender
356 </smp:ParticipantIdentifier>
357
```

358 This message is sent as a business message to the appropriate endpoint that has been resolved from the  
359 SML and SMP. This MAY be a LIME inbox for the original sender, or it MAY be a special destination  
360 managed by the sender's AP.

361 The XML is defined as follows.

362 **/ids:MessageUndeliverable**

363 The holding element is required. It has no attributes.

364 **/ids:MessageUndeliverable/ids:MessageIdentifier**

365 This required element contains the message identifier of the undelivered message. It is exactly  
366 the same as the corresponding header of the message that was not delivered.

367 **/ids:MessageUndeliverable/ids:ReasonCode**

368 This required element contains one of the following values of type xs:string:

- 369 1. METADATA\_ERROR: This string indicates that there was an addressing error with the  
370 message and no matching metadata can be found: for example the participant  
371 identifier is not available in the SML, or the document or process type is not listed in  
372 the SMP, or not known by the receiving Access Point.
- 373 2. TRANSPORT\_ERROR: This string indicates that the message cannot be delivered  
374 because of a failure to transport the document to the next hop in the chain. For  
375 example, the receiving AP's endpoint is not responding within a significant timeout, or  
376 there are faults in trying to connect, or the WSRM sequence is failing to complete.
- 377 3. RECIPIENT\_ERROR: There is a problem with the recipient. For example: The recipient  
378 has a full inbox; the recipient is uncontactable; the recipient's contract with the AP has  
379 expired or they have been suspended by the governance body.
- 380 4. SECURITY\_ERROR: This string indicates that there was an authentication error or  
381 security fault returned by the receiving AP, or that the receiver's certificate has been  
382 revoked.
- 383 5. OTHER\_ERROR: This string indicates that there was an error other than the above.

384 **/ids:MessageUndeliverable/ids:Details**

385 This required element of type xs:string contains human readable descriptions of the details of the  
386 problem. There is no restriction on its use. It MAY be empty.

387 **4.1.1 DocumentIdentifier**

388 The DocumentIdentifier of the metadata of a MessageUndeliverable message MUST be set to

```
389 <ids:DocumentIdentifier scheme="busdox-docid-qns">  
390     http://busdox.org/transport/lime/1.0/::MessageUndeliverable  
391 </ids:DocumentIdentifier>
```

392 **4.1.2 ProcessIdentifier**

393 The ProcessIdentifier of the metadata of a MessageUndeliverable message MUST be set to

```
394 <ids:ProcessIdentifier scheme="busdox-actorid-transport">  
395     busdox:noprocess  
396 </ids:ProcessIdentifier>
```

397

## 398 5 Appendix A

### 399 5.1 XML Schema for message identifiers

```
400 <?xml version="1.0" encoding="utf-8"?>
401 <xs:schema id="Identifiers"
402 targetNamespace="http://busdox.org/transport/identifiers/1.0/"
403 elementFormDefault="qualified"
404 xmlns="http://busdox.org/transport/identifiers/1.0/"
405 xmlns:xs="http://www.w3.org/2001/XMLSchema">
406
407 <xs:annotation>
408 <xs:documentation>
409 Common identifiers for WSDLs and Schemas
410 </xs:documentation>
411 </xs:annotation>
412
413 <xs:element name="ParticipantIdentifier"
414 type="ParticipantIdentifierType"/>
415 <xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
416 <xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>
417
418 <xs:element name="RecipientIdentifier"
419 type="ParticipantIdentifierType"/>
420 <xs:element name="SenderIdentifier" type="ParticipantIdentifierType"/>
421 <xs:element name="MessageIdentifier" type="MessageIdentifierType"/>
422 <xs:element name="ChannelIdentifier" type="ChannelIdentifierType"/>
423
424 <xs:complexType name="ParticipantIdentifierType">
425 <xs:simpleContent>
426 <xs:extension base="xs:string">
427 <xs:attribute name="scheme" type="xs:string" />
428 </xs:extension>
429 </xs:simpleContent>
430 </xs:complexType>
431
432 <xs:complexType name="DocumentIdentifierType">
433 <xs:simpleContent>
434 <xs:extension base="xs:string">
435 <xs:attribute name="scheme" type="xs:string" />
436 </xs:extension>
437 </xs:simpleContent>
438 </xs:complexType>
439
440 <xs:complexType name="ProcessIdentifierType">
441 <xs:simpleContent>
442 <xs:extension base="xs:string">
443 <xs:attribute name="scheme" type="xs:string" />
444 </xs:extension>
445 </xs:simpleContent>
446 </xs:complexType>
```

```
447
448 <xs:simpleType name="MessageIdentifierType">
449   <xs:restriction base="xs:string" />
450 </xs:simpleType>
451
452 <xs:simpleType name="ChannelIdentifierType">
453   <xs:restriction base="xs:string" />
454 </xs:simpleType>
455
456 </xs:schema>
```