



# Business Document Exchange Network - Common Definitions

---

*Version 0.9.5*



WP8 2009-10-21



## Contents

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 .....	6
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	BusinessIdentifiers.....	9
3.3.1	Universal Business Identifier Scheme.....	9
3.3.2	XML format for Business Identifiers .....	10
3.3.3	Using Business Identifiers in URLs .....	10
3.4	DocumentIdentifier .....	11
3.4.1	XML Representation of Document Identifiers.....	12
3.4.2	URL representation of Document Identifiers .....	12
3.5	Process Identifiers .....	13
3.5.1	XML Representation of Process Identifiers .....	13
3.6	BUSDOX defined identifiers.....	14
3.6.1	Recipient Business Identifier .....	14
3.6.2	Sender Business Identifier .....	15
3.6.3	Document Type Identifier.....	15
3.6.4	Process Type Identifier .....	15

3.6.5	Message Identifier .....	15
3.6.6	Channel Identifier .....	15
3.7	Basic Profile .....	15
3.8	SOAP 1.1 .....	16
3.9	Use of HTTP .....	16
3.10	WS-Addressing 1.0.....	16
3.10.1	The <wsa:To> Header .....	16
3.10.2	The <wsa:Action> Header.....	16
3.10.3	The <wsa:MessageIdentifier> Header.....	16
3.10.4	The <wsa:RelatesTo> Header .....	16
3.10.5	EndpointReferences .....	16
4	Appendix A.....	17
4.1	XML Schema for message identifiers .....	17

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.

## 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.

### 4 **2.1 Goals and non-goals**

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

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

### 16 **2.2 Terminology**

17 Terminology use applies to all BUSDOX Transport Infrastructure Specifications.

#### 18 **2.2.1 Indication of Requirement Levels**

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

#### 21 **2.2.2 Common terms**

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

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

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

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

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

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

32 Secure Trusted Asynchronous Reliable Transport/**START**: The Secure Trusted Asynchronous Reliable  
33 Transport.

34 Lightweight Message Exchange/**LIME**: The Lightweight Message Exchange Transport.

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

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

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

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

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

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

47 Service Metadata Locator service/**SML**: A service which provides a client with the capability of discovering  
48 the Service Metadata Publisher endpoint associated with a particular Business Identifier. A client uses this  
49 service in order to find where information is held about services for a particular business.

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

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

57 Business Identifier: A business level identifier such as GLN or DUNS number that is used to identify a trading  
58 partner. In the context of BUSDOX, business identifiers are used to discover services associated with trading  
59 partners in Service Metadata.

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

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

63 Service Metadata Publisher Certificate (**SMP Certificate**): A certificate used by a specific SMP to create all  
64 signatures of the signed resources. This certificate must correspond to the certificate referenced by the  
65 SML as being associated with the SMP endpoint.

66 **2.2.3 Notational conventions**

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

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

```
76  
77 <!-- sample pseudo-schema -->  
78 <defined_element  
79     required_attribute_of_type_string="xs:string"  
80     optional_attribute_of_type_int="xs:int"? >  
81 <required_element />  
82 <optional_element />?  
83 <one_or_more_of_these_elements />+  
84 [ <choice_1 /> | <choice_2 /> ]*  
85 </defined_element>
```

#### 86 **2.2.4 Normative references**

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

88 [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/>

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

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

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

#### 94 **2.2.5 Non-normative references**

95 [WS-T] "Web Services Transfer (WS-Transfer)", W3C Editor's Draft 22 April 2009,  
96 <http://www.w3.org/2002/ws/ra/edcopies/wst.html> (retrieved on 28/4/2009)

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

99

100

101 **2.3 Namespaces**

102 The following table lists XML namespaces that are used in this document. The choice of any namespace  
103 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>
wsu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>
ids	<a href="http://busdox.org/transport/identifiers/1.0/">http://busdox.org/transport/identifiers/1.0/</a>
ds	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

104

105

## 106 **3 Identifiers**

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

### 109 **3.1 Notational conventions**

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

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

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

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

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

115 `/{identifier}/service[/endpoint]`

116 Can be instantiated to either of the strings

117 `/0010:5798000000001/service`

118 And

119 `/0010:5798000000001/service/endpoint`

### 120 **3.2 On the use of percent encoding in URLs**

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

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

### 125 **3.3 BusinessIdentifiers**

126 Business identifiers logically consist of a *scheme identifier* and the *business identifier* itself. Business  
127 Identifiers are associated with groups of services, or Service Metadata.

128 The *scheme identifier* indicates the specification of the business identifier format, i.e. its representation and  
129 meaning. Currently there is only one scheme defined, the *Universal Business Identifier Scheme*, and is  
130 identified by the URI “<http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifier/1.0/>”.

#### 131 **3.3.1 Universal Business Identifier Scheme**

132 The default business identifier scheme in BUSDOX is the ‘universal business identifier scheme’. This is a  
133 transitory scheme for the early versions of the BUSDOX specifications, with the final scheme to be decided.

134 Currently, the scheme is {type identifier};{business identifier}.

135 The type identifier is 4-digit number indicating the type of business identifier, such as GLN, DUNS, CVR or  
136 another scheme. This scheme is dynamically extendable and will be decided outside of this specification.

137 Note that the {type identifier} part of the business identifier *is not* equivalent to the scheme of the  
 138 identifier - the {type identifier} indicates the *type of business identifier*, whereas the scheme identifier  
 139 indicates the format and semantics of an identifier string.

140 For the moment, only the type identifier “0010” has been defined as indicating a GLN number, such as for  
 141 example “0010:5798000000001”.

### 142 3.3.2 XML format for Business Identifiers

143 The <BusinessIdentifier> element is used to represent business identifiers and scheme information.

144 Pseudo-scheme for BusinessIdentifier:

```
145 <smp:BusinessIdentifier scheme="xs:anyURI">
146   xs:string
147 </smp:BusinessIdentifier>
```

148  
 149 Where the ‘scheme’ attribute indicates the scheme of the business identifier.

Field	Description
BusinessIdentifier	A business identifier which may be associated with a group of services.
BusinessIdentifier/ @scheme	<p>The scheme of the business identifier. This scheme indicates the <i>format</i> of the business identifier (i.e. the textual format) – not its semantic type (e.g. DUNS or GLN).</p> <p>This scheme type <b>MUST</b> be in the form of a URI.</p> <p>The URI that indicates the “Universal Business Identifier” format is            “http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifier/1.0/”</p> <p>When processing a business identifier in XML format, it <b>MUST</b> be treated as case insensitive.</p>

150

### 151 3.3.3 Using Business Identifiers in URLs

152 The following format is used:

153 {identifier scheme}::{id}

154 Where ‘identifier scheme’ is the format of the identifier, and ‘id’ is the business identifier itself, following  
 155 the format indicated by the ‘identifier type’ part. The {id} could for example be the format defined as  
 156 ‘Universal Business Identifier’ scheme.

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

159 Non-normative example that uses the Universal Business Identifier Format, assuming the business  
 160 identifier “0010:5798000000001”:

161 <http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifier/1.0/::0010:5798000000001>

162 In percent encoded form:

163 `http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fUniversalBusinessIdentifier%2f1.0%2f`  
164 `%3a%3a0010%3a579800000001`

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

### 168 **3.4 DocumentIdentifier**

169 Documents are represented by an identifier (identifying the document type) and a scheme type which  
170 represents the scheme or format of the identifier itself.

171 This document defines only one identifier scheme, the 'QName/Subtype Identifier Scheme', which is  
172 identified by the following URI:

173 `http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/`

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

176 `{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}]`.

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

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

- 179
- 180 • **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2
  - 181 • **Document element local name:** Order
  - 182 • **Subtype identifier:** UBL-2.0 (since several versions of the Order schema may use the same  
namespace + document element name)

183 The document type identifier will then be:

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

185

186 **3.4.1 XML Representation of Document Identifiers**

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

188 Pseudo-scheme for DocumentIdentifier:

189 `<DocumentIdentifier scheme="xs:anyURI">xs:string</DocumentIdentifier>`

190

191 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".
BusinessIdentifier/ @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>http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/</i>  When processing a document identifier in XML format, it MUST be treated as case insensitive.

192

193 **3.4.2 URL representation of Document Identifiers**

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

196 "http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/".

197 The format of this is:

198 {identifier scheme}::{id}

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

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

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

202 • **Identifier scheme:**

203 `http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/`

204 • **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2

205 • **Document element local name:** Order

206 • **Subtype identifier:** UBL-2.0 (since several versions of the Order schema may use the same  
207 namespace + document element name)

208 The document type identifier will then be:

209 `http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/::urn:oasis:nam`  
210 `es:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0`

211 Rules for parsing this identifier:

- 212 • The text up until the first “::” is the identifier scheme identifier
- 213 • The text before the next to last “::” and last “::” is the root namespace
- 214 • The text between the last occurrence of ‘::’ and last occurrence of ‘##’ OR end of the string is the  
215 document element local name.
- 216 • The text following the first ‘##’ after the document element local name (if any) is the subtype  
217 identifier.

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

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

222 `http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fDocumentIdentifierType%2fQNameSu`  
223 `btype%2f%3a%3aurn%3aoasis%3anames%3aspecification%3aubl%3aschema%3axsd%3aOrder-`  
224 `2%3a%3aOrder%23%23UBL-2.0`

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

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

## 231 **3.5 Process Identifiers**

232 A process identifier represents a process (UBL terminology: *Profile*) that a specific document type can  
233 participate in. Process identifiers consist of the process identifier itself, and a scheme or identifier format  
234 type.

### 235 **3.5.1 XML Representation of Process Identifiers**

236 Pseudo-schema for the ProcessIdentifier XML element:

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

238

239

240 Description of the individual fields (elements and attributes).

241

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.</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 may for example choose to use the identifier “http://busdox.org/processIdentifiers/1.0/UBL/CENBII/1.0/” to indicate the format of process identifiers.</p> <p>Although an URI, the process identifier scheme <b>MUST</b> be treated as case insensitive.</p>

242

243

### 244 3.6 BUSDOX defined identifiers

245 The identifiers defined in this section are used both with the Service Metadata specifications and the  
246 transport specifications. Every BUSDOX message has associated metadata included, in the form of headers,  
247 so that Access Points can route messages without looking inside the business message.

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

#### 249 3.6.1 Recipient Business Identifier

250 This element represents the business identifier of the ultimate recipient. This is used for Service Metadata  
251 Lookup and message forwarding. Pseudo schema for this element is:

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

253 This element contains both the Recipient identifier and identifier scheme. It follows the rules for XML  
254 representation of business identifiers laid out in section 3.3.

### 255 **3.6.2 Sender Business Identifier**

256 This element represents the business identifier of the original sender. This is used for Service Metadata  
257 lookup and message forwarding. Pseudo schema for this element is:

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

259 This element contains both the Sender identifier and identifier scheme. This element contains both the  
260 Recipient identifier and identifier type. It follows the rules for XML representation of business identifiers  
261 laid out in section 3.3.

### 262 **3.6.3 Document Type Identifier**

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

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

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

### 268 **3.6.4 Process Type Identifier**

269 This element represents the type of process that a document may participate in. This is used for Service  
270 Metadata lookup and message forwarding. Pseudo schema for this element is:

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

272 This element follows the rules for XML representation of process identifiers laid out in section 3.5.

### 273 **3.6.5 Message Identifier**

274 Because BUSDOX Messages may pass between several parties (for example in the "four-corner" model,  
275 from LIME client1 to AP1 to AP2 to LIME client2), it is necessary to have a constant message identifier that  
276 uniquely identifies the message across multiple hops. This message identifier is contained in the:

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

278 element.

279 If the message originated with a LIME client, then the message ID used to send the message on to the  
280 DestAP MUST be the same BUSDOX Message ID used in the LIME interaction. In the case that there is no  
281 suitable MessageIdentifier when the message originates at the SrcAP, then the SrcAP MUST allocate a  
282 unique identifier for this message. This MessageIdentifier MAY be the same as the wsa:MessageIdentifier.

### 283 **3.6.6 Channel Identifier**

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

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

## 287 **3.7 Basic Profile**

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

289 **3.8 SOAP 1.1**

290 APs MUST use SOAP 1.1 for all message exchange.

291 Messages MUST use the document/literal style.

292 **3.9 Use of HTTP**

293 HTTP messages MUST be exchanged over TCP/IP port 80. HTTPS messages MUST be Exchanged over TCP/IP  
294 port 443.

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

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

300 **3.10 WS-Addressing 1.0**

301 This profile only supports the use of WS-Addressing 1.0 and messages MUST use the namespace:

302 <http://www.w3.org/2005/08/addressing>

303 **3.10.1 The <wsa:To> Header**

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

305 **3.10.2 The <wsa:Action> Header**

306 All message interactions MUST include a wsa:Action header, and this MUST match the action value of the  
307 HTTP SOAPAction header when HTTP is the transport.

308 **3.10.3 The <wsa:MessageIdentifier> Header**

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

310 Any party that assigns a value to a <wsa:MessageIdentifier> header block MUST ensure that there is  
311 negligible probability that that party or any other party will accidentally assign the same identifier to any  
312 other message. Parties SHOULD use the UUID standard for generating message Ids.

313 Note that [WSAv1.0] requires that <wsa:MessageIdentifier> values be absolute IRIs.

314 **3.10.4 The <wsa:RelatesTo> Header**

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

317 **3.10.5 EndpointReferences**

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

319

320

## 321 4 Appendix A

### 322 4.1 XML Schema for message identifiers

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Identifiers"
  targetNamespace="http://busdox.org/transport/identifiers/1.0/"
  elementFormDefault="qualified"
  xmlns="http://busdox.org/transport/identifiers/1.0/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:annotation>
    <xs:documentation>
      Common identifiers for WSDLs and Schemas
    </xs:documentation>
  </xs:annotation>

  <xs:element name="BusinessIdentifier" type="BusinessIdentifierType"/>
  <xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
  <xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>

  <xs:element name="RecipientIdentifier" type="BusinessIdentifierType"/>
  <xs:element name="SenderIdentifier" type="BusinessIdentifierType"/>
  <xs:element name="MessageIdentifier" type="MessageIdentifierType"/>
  <xs:element name="ChannelIdentifier" type="ChannelIdentifierType"/>

  <xs:complexType name="BusinessIdentifierType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scheme" type="xs:anyURI" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="DocumentIdentifierType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scheme" type="xs:anyURI" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="ProcessIdentifierType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="scheme" type="xs:anyURI" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="MessageIdentifierType">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

  <xs:simpleType name="ChannelIdentifierType">
    <xs:restriction base="xs:string" />
  </xs:simpleType>

</xs:schema>
```