

1
2



European Network of
Transmission System Operators
for Electricity

ENTSO-E XML NAMESPACE REFERENCE DOCUMENT

2016-04-19
DOCUMENT APPROVED
VERSION 3.0

3

Table of Contents

4	1	Overview	5
5	2	Introduction	5
6	3	URI.....	5
7	4	targetNamespace	6
8	5	Electronic document instance files	7
9	6	File naming convention	8
10	7	ENTSO-E electronic document Conformity	9
11	7.1	ENTSO-E Schema additions to cater for Local variations	10
12	7.2	Core Component Schema	10
13	7.3	Code List Schema.....	11
14	7.4	Local code list extensions.	12
15	7.5	Local Document use overview.....	13
16	8	Annex 1: Modifications to the schemas defined before 2012	14
17	8.1	Etso-code-lists.xsd.....	14
18	8.2	Etso-core-cmpts.xsd	14
19	8.3	Electronic document Schemas	14
20	9	Annex 2: Latest version of entso-e schemas	15
21		List of figures	
22		Figure 1 - XML Schema extract.....	7
23		Figure 2 - Instance document respecting the XML Schema in Figure 1	7
24		Figure 3 - Recommended structure of an instance document	8
25		Figure 4 - ENTSO-E Compliant schema	9
26		Figure 5 - ENTSO-E Document extensions	10
27		Figure 6 - ENTSO-E Codelist structure	11
28		Figure 7 - Overall picture of local code use	13
29			

30

Copyright notice:

31 **Copyright © ENTSO-E. All Rights Reserved.**

32 This document and its whole translations may be copied and furnished to others, and derivative
33 works that comment on or otherwise explain it or assist in its implementation may be prepared,
34 copied, published and distributed, in whole or in part, without restriction of any kind, provided
35 that the above copyright notice and this paragraph are included on all such copies and
36 derivative works. However, this document itself may not be modified in any way, except for
37 literal and whole translation into languages other than English and under all circumstances, the
38 copyright notice or references to ENTSO-E may not be removed.

39 This document and the information contained herein is provided on an "as is" basis.

40 **ENTSO-E DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT**
41 **LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT**
42 **INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR**
43 **FITNESS FOR A PARTICULAR PURPOSE.**

44

Maintenance notice:

45 This document is maintained by the ENTSO-E WG EDI. Comments or remarks are to be
46 provided at EDI.Library@entsoe.eu

47

Revision History

Version	Release	Date	Comments
1	0	2011-03-16	First version of the document. Document approval by the Market Committee
2	0	2012-10-11	New harmonized approach for defining the namespace as well as the management of extension/restriction of code list. Document approved by the Market Committee on 2012-10-23.
3	0	2016-04-19	Cancellation of the use of the "restricted codelist". Thus all the codes defined in the ENTSO-E codelist or added in the "local extension" could be used in an exchange if this is agreed between the parties. "VersionRelease" referring to the version of the ENTSO-E core component has been removed as the ENTSO-E core component are frozen. This is as per decision of ENTSO-E to use standards based on CIM and published as European norm (IEC/EN norms). Added note in the overview chapter stating that ENTSO-E schemas based on ENTSO-E components are frozen.

48

49 1 Overview

50 XML namespace is the method to qualify element and attribute names used in XML Schema
51 through their association with a unique reference that is identified by a URI (Uniform Resource
52 Identifier) reference.

53 XML namespace use is defined in the W3C document “Namespaces in XML” (Third Edition) and
54 can be located at the following URL: <http://www.w3.org/TR/xml-names/>

55 URIs are defined in the document RFC (Request For Comments) 3986 published by the IETF
56 (Internet Engineering Task Force) and can be located at the following URL: [http://www.rfc-
57 editor.org/rfc/rfc3986.txt](http://www.rfc-editor.org/rfc/rfc3986.txt)

58 **Please note that as ENTSO-E's WG EDI work products are standardised in the IEC and**
59 **reflected in the IEC 62325 series, the International Standard IEC 62325-450 "Profile and**
60 **context modelling rules" has become the reference for modelling purposes, methodology**
61 **wise. Thus, ENTSO-E schemas based on ENTSO-E components are frozen. XSLT**
62 **(EXtensible Stylesheet Language) to convert ENTSO-E schemas into instances based on**
63 **CIM are available on ENTSO-E web site.**

64 2 Introduction

65 With the introduction of the possibility of extend ENTSO-E code lists in addition to the increasing
66 use of web services and its requirement for WSDL (Web Services Description Language),
67 ENTSO-E WG EDI agreed that time has come to introduce namespace use within the ENTSO-
68 E XML Schema.

69 The principle objective of these namespaces will be to identify each electronic document in a
70 manner that ensures its uniqueness.

71 An XML namespace is identified by a URI reference. All schema element and attribute names
72 may be placed in an XML namespace.

73 The element name taken in isolation (for example “DocumentIdentification”) is known as a local
74 name. However, in order to completely understand the meaning behind a local name it is
75 necessary to know to which namespace (i.e. environment where the word is used) it belongs.
76 The combination of both names (i.e. the local name and namespace name) is known as an
77 expanded name¹.

78 To understand an XML element name it is necessary to prefix the namespace name to the local
79 name (for example, “ScheduleDocument:DocumentIdentification”; where “ScheduleDocument”
80 corresponds to the namespace name and “DocumentIdentification” corresponds to the element
81 name). However, since namespace names can be very long and can even contain characters
82 not allowed in names, a “prefix” is used in their place to represent them. The combination of
83 the prefix and the local name is known as a qualified name².

84 The prefix, usually of three characters, appears in the place of the namespace name (for
85 example, ess:DocumentIdentification where “ess” has been associated with the namespace
86 “ScheduleDocument”).

87 The use of namespaces may also allow a TSO to tailor a specific codelist to a specific set of
88 exchanges. Consequently the TSO has the possibility to give different meanings to the internal
89 code “Z01” in a codelist for different information exchanges.

90 3 URI

91 A Uniform Resource Identifier (URI) provides a simple and extensible means for identifying a
92 resource. The term "resource" is used here in a general sense for whatever might be identified
93 by a URI. In the case of ENTSO-E it will be used to identify the different contexts in which XML

¹ Definition: An **expanded name** is a pair consisting of a [namespace name](#) and a [local name](#).

² Definition: A **qualified name** is a name subject to namespace interpretation.

94 names are declared. This, more specifically enables the definition of a namespace for each of
95 the ENTSO-E processes.

96 It is important to note that the URI only provides identification of a resource; access to the
97 resource is neither guaranteed nor is it implied by the presence of a URI.

98 In order to provide a generic and stable means of declaring a URI for ENTSO-E it will be
99 composed in the following manner:

100 **urn:entsoe.eu:<ENTSO-E Domain>[:<process>][:<document>][:<Version>[:<Release>]]**

101 where:

- 102 – **urn:entsoe.eu** shall be the stem of all ENTSO-E namespaces.
- 103 – **<ENTSO-E Domain>** identifies the organisation or group of organisations within ENTSO-E
104 that owns the object being referenced. In the case of WG EDI this shall be “wgedi”. In the
105 case of a TSO it could be “ree”, “rte”, “ceps”, “seps”, etc.
- 106 – **<process>** when possible identifies the specific process within the ENTSO-E Domain where
107 the object is situated. In the case of WG EDI this shall be the processes for which electronic
108 documents have been defined (for example, ess, ecan, errp, etc.).
- 109 – **<document>** optionally identifies the electronic document.
- 110 – **<Version>** optionally identifies the version of the document.
- 111 – **<Release>** optionally identifies the release of the document.

112 Examples of specific instances of ENTSO-E URIs:

- 113 • urn:entsoe.eu:wgedi:acknowledgement:acknowledgementdocument:6:1
- 114 • urn:entsoe.eu:wgedi:errp:activationdocument:4:0
- 115 • urn:entsoe.eu:wgedi:esp:energyaccountreport:1:1
- 116 • urn:entsoe.eu:wgedi:ess:schedulingimplementationguide:4:0
- 117 • urn:entsoe.eu:wgedi:components
- 118 • urn:entsoe.eu:wgedi:codelists
- 119 • etc.

120 It is to be noted that for simplicity and to avoid naming conflicts all the characters used in a
121 namespace shall be in lower case. A namespace name is case sensitive and this will avoid
122 problems due to the wrong use of a character case.

123 **4 targetNamespace**

124 Within an XML Schema a targetNamespace is used to specify to which namespace all the
125 schema constructs belong (e.g. element declarations, attribute declaration and type definitions).
126 This in fact enables one to define the namespace of the schema that can be used for validation.

127 The targetNamespace attribute can be used to place the elements and attributes of the schema
128 into a specific namespace. This ensures that their use and meaning is qualified by the schema
129 in which they belong.

130 The default XML namespace will be used within the ENTSO-E XML schema to ensure that all
131 the elements defined in the schema document are restricted to it by default. Consequently they
132 will not require a specific namespace prefix.

133 For example in the ScheduleDocument XML schema the information will be as presented as
134 outlined in Figure 1.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:ecc="urn:entsoe.eu:wgedi:components"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2"
  targetNamespace="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xsd:import namespace="urn:entsoe.eu:wgedi:components" schemaLocation="../../../core/urn-entsoe-eu-wgedi-components.xsd"/>
  <xsd:element name="ScheduleDocument">
    <xsd:complexType>
      <xsd:annotation>
        <xsd:documentation/>
      </xsd:annotation>
      <xsd:sequence>
        <xsd:element name="DocumentIdentification" type="ecc:IdentificationType">
          <xsd:annotation>
            <xsd:documentation/>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="DocumentVersion" type="ecc:VersionType">
          <xsd:annotation>
            <xsd:documentation/>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="ScheduleTimeSeries" type="ScheduleTimeSeries_Type" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

```

Callouts in the image:

- The namespace defining the URI of the core components (points to `xmlns:ecc`)
- The namespace defining the URI of the XML Schema components (points to `xmlns:xsd`)
- The default namespace defining the URI of specific document components (points to `xmlns`)
- The attribute defining the URI to be used for the components of this document (points to `targetNamespace`)
- The attribute indicating where the core components may be found (points to `schemaLocation`)
- This is to be found in the XML schema namespace (points to `xmlns`)
- This is to be found in the core components (points to `ecc:IdentificationType`)
- This is to be found in the document itself (points to `DocumentIdentification`)

135

136

Figure 1 - XML Schema extract

137 The default XML namespace may be used within the ENTSO-E XML instance documents.
138 Elements belonging to that namespace would not require a specific namespace prefix.

139 The introduction of a Namespace slightly changes the beginning of the XML instance document
140 to take the new attributes requirements into account. An example of such an instance document
141 is shown in Figure 2.

```

<?xml version="1.0" encoding="UTF-8"?>
<ScheduleDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2 urn-entsoe-eu-wgedi-ess-scheduledocument-4-2.xsd"
  xmlns="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2">
  <DocumentIdentification v="a"/>
  <DocumentVersion v="1"/>
  <DocumentType v="A94"/>
  <ProcessType v="A39"/>
  <ScheduleClassificationType v="A01"/>

```

Callouts in the image:

- Hint regarding the location of schema relevant for validation (points to `xsi:schemaLocation`)
- Identification of the default namespace of the document (points to `xmlns`)

142

143

Figure 2 - Instance document respecting the XML Schema in Figure 1

144 The XML instance header information changes in two basic manners:

- 145 a) The optional introduction of the attribute "schemaLocation" to provide a hint of where the
146 XML Schema file that can be used to validate xml elements associated with the provided
147 namespace can be found.
- 148 b) The introduction of a default namespace definition to indicate that elements without a prefix
149 belong to the provided namespace.

150 5 Electronic document instance files

151 In the creation of an instance document that is strictly compliant with the XML Schema it is
152 necessary to identify the namespace being used, generally through the use of the default
153 namespace in the heading of the instance document as indicated in Figure 2.

154 However, prefixes should not be used for the elements since a default ENTSO-E namespace is
155 provided.

156 Prefixes could be used for local purposes not explicitly described in ENTSO-E namespace
157 recommendation. In this case an agreement is required.

158 It may also be helpful to identify the location of the schema with the xsi:schemaLocation
159 instruction as can also be seen in Figure 2.

160 The use of the xsi:schemaLocation generally provides the receiver of the document with a “hint”
161 of where the schema for a specific namespace itself may be found. For example the instruction
162 as shown in Figure 2:

163 `xsi:schemaLocation="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2 urn-entsoe-eu-wgedi-ess-`
164 `scheduledocument-4-2.xsd"`

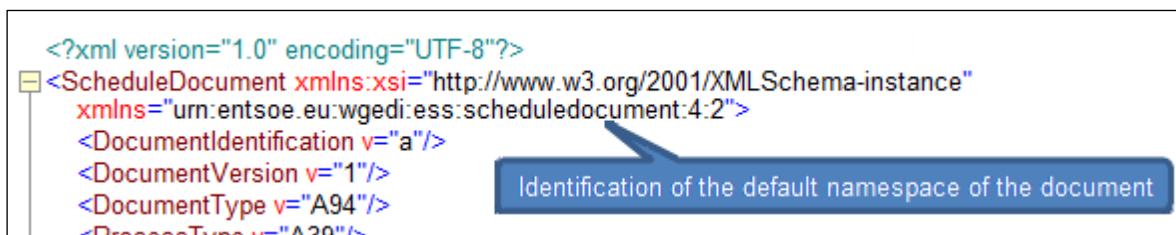
165 Implies that the schema itself may perhaps be found at:

166 `“urn-entsoe-eu-wgedi-ess-scheduledocument-4-2.xsd”`.

167 However this is not always the case and in general an agreement is required.

168 **In order to enable flexibility, it is recommended that the schema location instruction (and**
169 **xsi definition) in the schema compliant instance should not be used.**

170 Removing such instructions from the instance document results in an XML document that does
171 not specifically identify a location for the XML Schema to which it is compliant. It is consequently
172 up to the receiving party to use the appropriate schema that defines the provided namespace
173 to validate the information instance.



174

175 **Figure 3 - Recommended structure of an instance document**

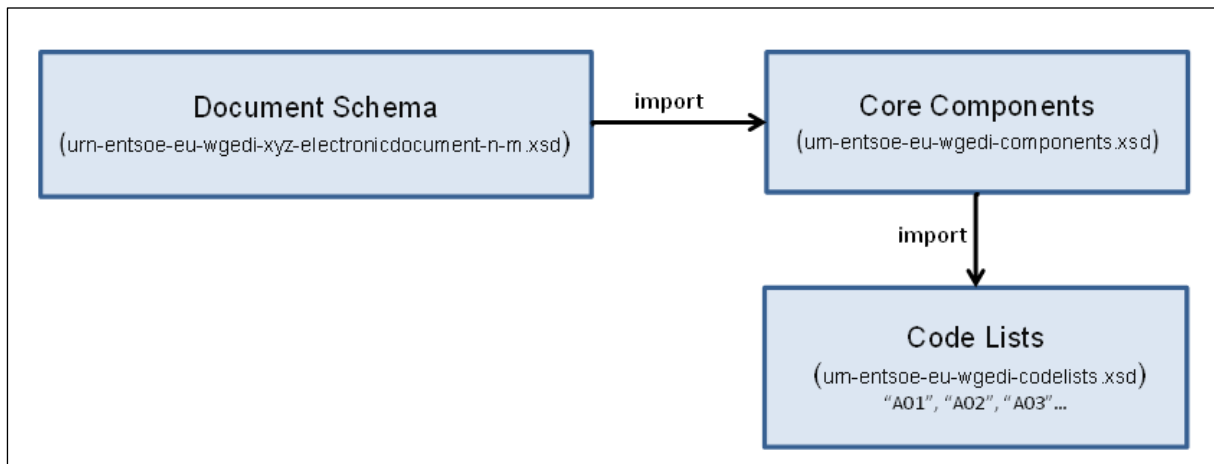
176 **6 File naming convention**

177 In order to facilitate in a consistent manner the way files containing schemas should be named
178 it is recommended to make use of the URI that is defined for the schema in question. However
179 since the colon (":") cannot be used in a filename it is recommended to replace each colon and
180 dot (".") with a dash ("-").

181 Consequently the file name for the schema with the URI
182 "urn:entsoe.eu:wgedi:ess:scheduledocument:4:2" would be "urn-entsoe-eu-wgedi-ess-scheduledocument-4-
183 2.xsd".

184 **7 ENTSO-E electronic document Conformity**

185



186

187
188

Note: the term “xyz-electronicdocument-n-m” should be replaced by the specific process and name, version and release of the document in question.

189

Figure 4 - ENTSO-E Compliant schema

190 The standard ENTSO-E electronic document XML schema is composed of the following
191 standardized elements as outlined in Figure 4:

192 a) The “Document Schema” itself that has been automatically generated from the UML model
193 as described in the implementation guide. The electronic document XML schema file is
194 named “urn-entsoe-eu-wgedi-xyz-electronicdocument-n-m.xsd” where “xyz-
195 electronicdocument” corresponds to the process and name of the electronic document and
196 “n-m” corresponds to the version and release of the document in question.

197 b) The “Document Schema” imports the basic set of ENTSO-E “Core Components” in order to
198 provide the complete set of XML datatypes that are required within the ENTSO-E
199 environment. This Schema is a generic schema that is valid for all ENTSO-E electronic
200 document schemas. The core component xml schema file is named “urn-entsoe-wgedi-
201 components.xsd”.

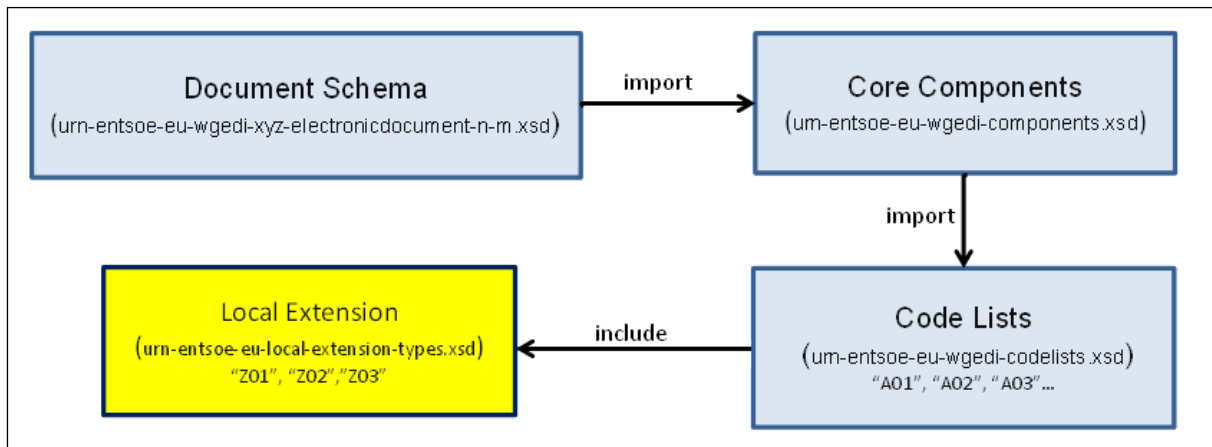
202 c) The “Core Components” schema itself imports the set of codelists that have been approved
203 within ENTSO-E. This Schema is a generic schema that is valid for all ENTSO-E electronic
204 document schemas. The code list xml schema file is named “urn-entsoe-eu-wgedi-
205 codelists.xsd”.

206 The Document Schema defines an electronic document that has been clearly documented
207 within an ENTSO-E process (ESS, ESP, ERRP, etc.).

208 An electronic document instance shall be considered ENTSO-E compliant if it respects the
209 information provided by the above “Document Schema”.

210 **7.1 ENTSO-E Schema additions to cater for Local variations**

211



212

213 **Figure 5 - ENTSO-E Document extensions**

214 An ENTSO-E local variation consists of an ENTSO-E provided schema with additional new valid
215 local codes to satisfy local market needs. The specific local codes are defined in the Local
216 extension. No other changes are permitted.

217 **7.2 Core Component Schema**

218 The Core Component Schema defines all the data types that are used within the ENTSO-E
219 environment. No ENTSO-E electronic document can be generated using data types that are not
220 referenced in the Core Component Schema. This ensures that all ENTSO-E electronic
221 documents make use of a consistent set of information blocks.

222 The Core Component Schema is divided into two parts:

- 223 a) A set of unqualified data types that have no specific constraints applied to the information
224 content other than the basic structural rules within their definition.
- 225 b) A set of qualified data types that represent the complete set of code lists defined for use
226 within ENTSO-E electronic documents. The codelists are in general of a qualifying nature
227 and provide additional consistent semantics to the electronic document content. These
228 codelists are defined in the Standard ENTSO-E codelist set package.

229 The Core Component Schema along with the ENTSO-E standard codelist set comprise the
230 basic building blocks necessary for the creation of all ENTSO-E electronic documents.

231

232

233

234

235

236

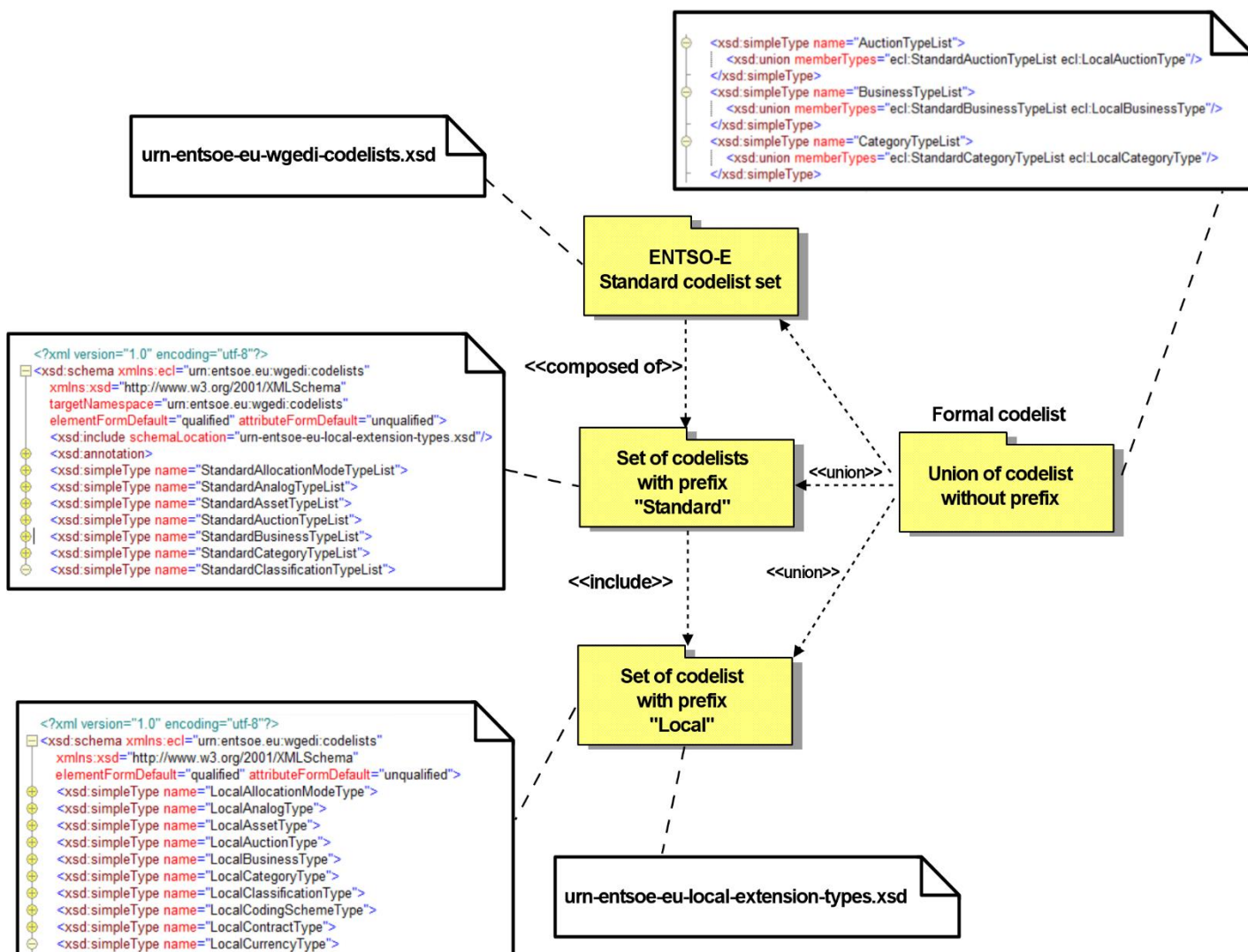
237

238

239

240

241 7.3 Code List Schema



242 Figure 6 - ENTSO-E Codelist structure

243 As can be seen in **Erreur ! Source du renvoi introuvable.** the standard codelist set is itself
244 divided into two basic parts:

- 245 a) The list of “standard” codelists that consists of all the codes that have been approved within
246 WG-EDI.
- 247 b) A list of “local” codelists that are assigned by individual System Operators for internal local
248 market use. These codes are local to a given market and are managed by the local System
249 Operator in order to satisfy local market constraints. These codes cannot be used outside
250 the market in which they have been designed.

251 In order to provide a stable core component library, ENTSO-E makes available the library
252 structure that includes an “empty” local set of codelists. This “empty” set of codelists in fact
253 contains by default the first “standard” code that can be found within the “standard” codelist.
254 This ensures that within the value space of the “local” codelist there is no possibility of validating
255 codes that do not exist. This local codelist schema has a standardised name within the ENTSO-
256 E environment which is “urn-entsoe-eu-local-extension-types.xsd”.

257 Each System Operator then replaces the “empty” local set of codelists with the “local” codelists
258 that are needed within its local marketplace. It is also the responsibility of the System Operator
259 to ensure the distribution of the “local” codelist to the Market Participants within the local market
260 area.

261 The ENTSO-E XML Schema environment “merges” the two codelist sets in order to provide at
262 the electronic document level the complete set of codes possible (i.e. the union of the “standard”
263 and “local” codes).

264 **7.4 Local code list extensions.**

265 The “Code List” XML Schema is composed of all the codes for each individual ENTSO-E codelist
266 (BusinessTypes, ProcessTypes, etc). Within the XML Schema, each of the codelists is named
267 with the prefix “Standard” (StandardBusinessType, StandardProcessType, etc.).

268 The “Code List” Schema also includes an XML Schema with an equivalent set of local codelists
269 that may be defined for specific local use. Within the local schema they have the prefix “Local”
270 (LocalBusinessType, LocalProcessType, etc.).

271 The standard “Code List” XML schema has in addition one other entry that provides the junction
272 of the standard and local codelists. The name of the codelist has no prefix (BusinessType,
273 ProcessType, etc.) and it merely consists of the union of the local and the standard codelists
274 (e.g. StandardBusinessType, LocalBusinessType).

275 The complete codelist is always considered to be the union of the local and the standard
276 codelist.

277 ENTSO-E shall always provide a local code list under the name “urn-entsoe-eu-local-extension-
278 types.xsd” that contains all the codelists as found in the standard file, with the prefix “Local”
279 (LocalBusinessType, LocalProcessType, etc.).

280 By default, every local entry has a single enumeration that corresponds to the first entry that is
281 found in the standard codelist.

282 If a local code is to be defined for the local entry, the default entry shall be replaced by the local
283 entry. It is local market responsibility to maintain this electronic document and to ensure that it
284 remains up to date.

285 A local extension of the code list would require the replacement of one standard entry:

```
286 <xsd:simpleType name="LocalAllocationModeType">
287 <xsd:restriction base="xsd:NMTOKEN">
288 <xsd:minLength value="3"/>
289 <xsd:maxLength value="3"/>
290 <xsd:enumeration value="A01"/>
291 </xsd:restriction>
292 </xsd:simpleType>
```

293 By the local requirement:

```
294 <xsd:simpleType name="LocalAllocationModeType">
295 <xsd:restriction base="xsd:NMTOKEN">
296 <xsd:minLength value="3"/>
297 <xsd:maxLength value="3"/>
298 <xsd:enumeration value="Z01"/>
299 <xsd:enumeration value="Z02"/>
300 </xsd:restriction>
301 </xsd:simpleType>
```

302 Every time the ENTSO-E codelist is updated it will be necessary to replace the standard “urn-
303 entsoe-eu-local-extension-types.xsd” by the locally managed one to ensure that the local codes
304 are integrated into the extended codelist.

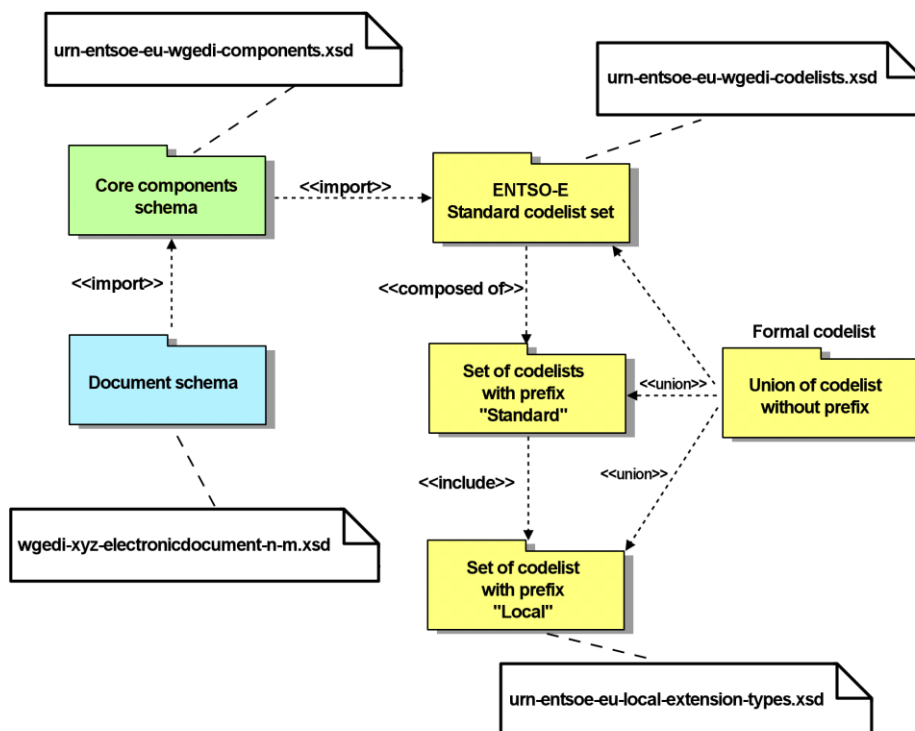
305

306

307

308

309 **7.5 Local Document use overview**



Note: The term "xyz-electronicdocument" should be replaced by the specific process and name of the document in question

Figure 7 - Overall picture of local code use

310
311
312 The diagram in Figure 7 provides the overall picture of the use of standard and local codelists
313 within the ENTSO-E environment. Where there are XML schema files the standardised format
314 of the filenames has been highlighted.

315 **8 Annex 1: Modifications to the schemas defined before 2012**

316 XML Schemas developed prior to the introduction of namespaces will require the changes as
317 outlined in the following paragraphs.

318 **8.1 Etso-code-lists.xsd**

319 The name of the xsd file will be changed to "urn-entsoe-eu-wgedi-codelists.xsd" to bring it into
320 line with the file naming convention as outlined in section 6.

321 There are no other changes other than the modification of the specific core component
322 namespace which is within the codelist schema.

323 **8.2 Etso-core-cmpts.xsd**

324 The name of the xsd file will be changed to "urn-entsoe-eu-wgedi-components.xsd" to bring it
325 into line with the file naming convention as outlined in section 6.

326 There are no other changes other than the modification of the specific core component
327 namespace.

328 **8.3 Electronic document Schemas**

329 The only change required is to the schema heading as follows:

```
330 <xsd:schema
331     xmlns:ecc="urn:entsoe.eu:wgedi:components"
332     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
333     xmlns="urn:entsoe.eu:wgedi:ess:scheduledocument:4:0"
334     targetNamespace="urn:entsoe.eu:wgedi:ess:scheduledocument:4:0"
335     elementFormDefault="qualified"
336     attributeFormDefault="unqualified"
337     ecc:VersionRelease="19.0">
338     <xsd:import namespace=
339 "urn:entsoe.eu:wgedi:components" schemaLocation="../../urn-entsoe-eu-wgedi-
340 components.xsd"/>
341     <xsd:include schemaLocation="urn-entsoe-eu-wgedi-scheduledocument-4-0-
342 restricted-codes.xsd"/>
```

343 The schema heading will contain the targetNamespace and with it the URI identifying the
344 electronic document. It also requires the introduction of a default namespace
345 (`xmlns="urn:entsoe.eu:wgedi:ess:scheduledocument:4:0"`) in order to ensure the connection with the
346 elements that are specific to the XML Schema in question. In addition the core component
347 namespace is aligned with the new namespace specification.

348 There are no other changes required to the electronic document schemas.

349 It should be noted that the introduction of a default namespace in the instance document that
350 corresponds to the default namespace in the XML schema will also ensure that a specific
351 namespace prefix is not required. This does not preclude its use if necessary and is agreed
352 between the parties.

353 **9 Annex 2: Latest version of entso-e schemas**

354 In the latest version of the entso-e schemas the following actions have been taken place:

- 355
- 356 • Cancellation of the use of the "restricted codelist". Thus all the codes defined in the ENTSO-
357 E codelist or added in the "local extension" could be used in an exchange if this is agreed
358 between the parties.
 - 359 • "VersionRelease" referring to the version of the ENTSO-E core component has been
360 removed as the ENTSO-E core component are freezed. This is as per decision of ENTSO-
361 E to use standards based on CIM and published as European norm (IEC/EN norms).

362

363 These changes are reflected in the following fragment using bold text:

```
364
365 <xsd:schema
366     xmlns:ecc="urn:entsoe.eu:wgedi:components"
367     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
368     xmlns="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2"
369     targetNamespace="urn:entsoe.eu:wgedi:ess:scheduledocument:4:2"
370     elementFormDefault="qualified"
371     attributeFormDefault="unqualified"
372     ecc:VersionRelease="19.0">
373     <xsd:import namespace=
374 "urn:entsoe.eu:wgedi:components" schemaLocation="../../urn-entsoe-eu-wgedi-
375 components.xsd"/>
376 <xsd:include schemaLocation="urn-entsoe-eu-wgedi-scheduledocument-4-0-restricted-
377 codes.xsd"/>

```

378

379