



1

19 June 2014

2

LEI Data File Format 1.0

3

4 **Abstract**

5 As the Global LEI System (GLEIS) High Level Principles stipulate, the GLEIS should uniquely
6 and unambiguously identify participants to financial transactions. The ISO 17442 standard defines a
7 set of attributes that are the most essential elements of identification, but this structure alone is not
8 sufficient. First, the semantic content of those attributes must be fully specified. Second, some
9 additional elements, such as an indication of the status of the information, are necessary for effective
10 use of the data. Third, the form the information takes at any given pre-LOU must be such that it can
11 be made to conform to a common standard, which must also be specified. This document proposes
12 the additional standards necessary in these areas to support the Interim GLEIS, particularly in
13 maintaining exclusive assignment of LEIs (one LEI per entity) and identifying, remediating data
14 quality issues, and supporting use of the data.

15
16 After approval of the Interim Standards by the ROC and after due time for adjustments by pre-LOUs,
17 all pre-LOUs will use this file format to publish pre-LEIs and their reference data. In the very
18 short term after ROC approval of the interim standards, pre-LOUs should prioritize any changes
19 to elements of their published data necessary for unambiguous mapping of the published data to
20 the common standard.

21 **Audience for this document**

22 The target audience for this interim standard includes:

- 23 • All pre-Local Operating Units (pre-LOUs) of the interim GLEIS
- 24 • All users or potential users of pre-LEI data from pre-LOUs
- 25 • All financial regulators who consume LEI data

26 **Status of this document**

27 This section describes the status of this document at the time of its publication. Other documents
28 may supersede this document.

29 **Table of Contents**

30 **1 Introduction.....5**

31 **1.1 Assumptions and Constraints.....6**

32 **2 Terminology and Typographical Conventions7**

33 **3 Abstract Data Content.....7**

34 **3.1 LEI File Header.....7**

35 **3.2 FileContentEnum Code List.....8**

36 **3.3 LEI Data Record.....9**

37 **3.3.1 Entity Section of LEI Data Record9**

38 **3.3.2 Registration Section of LEI Data Record12**

39 **3.3.3 Extension Section of LEI Data record13**

40 **3.4 Data Types14**

41 **3.4.1 Address Data Type14**

42 **3.4.2 AssociatedEntity Data Type.....14**

43 **3.4.3 BusinessRegisterEntityID Data Type15**

44 **3.4.4 CountryCode Data Type16**

45 **3.4.5 DateTime Data Type16**

46 **3.4.6 JurisdictionCode Data Type17**

47 **3.4.7 LanguageCode Data Type.....17**

48 **3.4.8 LEI Data Type.....17**

49 **3.4.9 Name Data Type.....18**

50 **3.4.10 OtherAddress Data Type18**

51 **3.4.11 OtherEntityName Data Type18**

52 **3.4.12 RegionCode Data Type19**

53 **3.4.13 SuccessorEntity Data Type19**

54 **3.5 Enumerated Code Lists.....19**

55 **3.5.1 AddressTypeEnum Code List.....20**

56 **3.5.2 AssociatedEntityTypeEnum Code List20**

57	3.5.3 BusinessRegisterEnum Code List	20
58	3.5.4 EntityExpirationReasonEnum Code List	20
59	3.5.5 EntityNameTypeEnum Code List.....	21
60	3.5.6 EntityStatusEnum Code List.....	21
61	3.5.7 RegistrationStatusEnum Code List.....	22
62	3.5.8 ValidationSourcesEnum Code List	26
63	4 Constraints and Data Validation.....	26
64	5 XML Syntax	27
65	5.1 XML Design Rules	27
66	5.2 XML Schema	27
67	6 Change Management.....	32
68	6.1 Minor Version Changes to the XML Schema	33
69	6.2 Major Version Changes to the XML Schema	35
70	7 Examples (non-normative).....	35
71	8 References	36
72	9 Appendix: Character Codes Allowed in ASCII Transliterated	
73	Names.....	37
74	10 Appendix: LEI Record Transition Diagram.....	39
75	11 Appendix: Considerations Regarding RegistrationStatus (non-	
76	normative)	40
77	11.1 Normal Lifecycle	41
78	11.2 Changes in Registration Status When the Status of the Legal	
79	Entity Changes	41
80	11.3 Changes in Registration Status When a Registration error is	
81	Resolved	42
82	11.4 Registration Status Codes for Transient Purposes.....	42
83		

84 **Introduction**

85 As the Global LEI System (GLEIS) High Level Principles stipulates, the GLEIS should uniquely
86 and unambiguously identify participants to financial transactions. The ISO 17442 standard
87 defines a set of attributes that are the most essential elements of identification, but this structure
88 alone is not sufficient. First, the semantic content of those attributes must be fully specified.
89 Second, some additional elements, such an indication of the status of the information, are
90 necessary for effective use of the data. Third, the form the information takes at any given pre-
91 LOU must be such that it can be made to conform to a common standard, which must also be
92 specified. This document proposes the additional standards necessary in these areas to support
93 the Interim GLEIS, particularly in maintaining exclusive assignment of LEIs (one LEI per entity)
94 and identifying, remediating data quality issues, and supporting use of the data.

95 After approval of the Interim Standards by the ROC and after due time for adjustments by pre-
96 LOUs, all pre-LOUs will use this file format to publish pre-LEIs and their reference data. In the
97 very short term after ROC approval of the interim standards, pre-LOUs should prioritize any
98 changes to elements of their published data necessary for unambiguous mapping of the published
99 data to the common standard.

100 This document specifies a standard file format for pre-LEI reference data. As the Global LEI
101 System High Level Principles stipulates, the Global LEI system should uniquely identify
102 participants to financial transactions. Therefore, it is critically important to eliminate a potential
103 cause of duplicates to meet the requirements of the global regulatory community for accurate,
104 consistent and unique entity identification. The standard set by this document is expected to
105 reduce risk of duplicates stemming from differences in publication formats used by pre-LOUs,
106 and to facilitate error detection to ensure data quality of the interim pre-LEI system. The standard
107 is expected to be used as a format for publishing reference data consolidated from all pre-LOUs,
108 for publishing reference data for an arbitrary subset of pre-LEIs, and so forth. It is also expected
109 the pre-LOUs which are currently publishing in their proprietary formats will adapt the standard
110 accordingly. Hence, all pre-LOUs will use this file format to publish pre-LEIs and their reference
111 data eventually.

112 The contents of this document are as follows:

- 113 • Section 2 defines terminology and typographical conventions.
- 114 • Section 3 specifies the abstract content of an LEI data file conforming to this standard,
115 including a detailed description of each element of reference data associated with a pre-LEI.
116 It also specifies the allowable values for data elements that are code lists.
- 117 • Section 4 specifies data validation constraints and other considerations intended to lead to
118 high-quality data content.
- 119 • Section 5 specifies a concrete realization of the data definitions above in XML syntax, by
120 means of XML schema (XSD 1.0).

- 121 • Section 6 specifies how this file format may be changed in the future, providing for
122 versioning, forward- and backward-compatibility, etc.
- 123 • Section 7 provides examples to illustrate the file format.
- 124 • Section 8 lists documents referenced herein.

125 **1.1 Assumptions and Constraints**

126 The following assumptions have guided the content of this interim standard:

- 127 • Structured data is needed to accurately transmit data between the publishers and the audience
128 of the LEI common file format.
- 129 • The file specification must account for and support the global implementation of the LEI
130 system.
- 131 • The file format is designed for public consumption, and the audience may include financial
132 industry participants, regulatory bodies, other pre-LOUs, the COU, data vendors, or other
133 interested parties
- 134 • The file will be published for use by pre-LOUs, and may be utilized for file publications by
135 the COU once the entity has become operational.
- 136 • Globally endorsed pre-LOUs will utilize this file format to publish their dataset of globally
137 accepted pre-LEI records.
- 138 • Although not currently required, pre-LOUs may also utilize this file format for standard
139 communications with other pre-LOUs. Specific use cases considered include cross-LOU
140 communication of a record transfer as well as the notification of a duplicate. In this context,
141 a broader expanse of the lifecycle of a registration may be covered, including a record that
142 has yet to receive an identifier.
- 143 • Implementation best practices will be utilized when possible.
- 144 • The resulting data file must take into account future expansion and modification
145 considerations as the GLEIS continues to evolve.

146 The following constraints have been considered:

- 147 • Information is already being published by pre-LOUs. As a result, consideration must be paid
148 to how existing entity records will be modified to account for this format.
- 149 • The audience for this file format may not be in direct contact with any implementation body.
150 As such, all possible uses of this information may not be accounted for.
- 151 • Information is already being consumed by stakeholders of the GLEIS. As a result,
152 consideration must be paid in how users will be migrated to the new file format.

- The file format references and honors previous completed work published by the ROC in a document entitled "A Common Data Format for Pre-LOUs to Use for Sharing or Publication of Data, part 1".

2 Terminology and Typographical Conventions

Within this specification, the terms SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY, NEED NOT, CAN, and CANNOT are to be interpreted as specified in Annex G of the ISO/IEC Directives, Part 2, 2001, 4th edition [ISODir2]. When used in this way, these terms will always be shown in ALL CAPS; when these words appear in ordinary typeface they are intended to have their ordinary English meaning.

All sections of this document, with the exception of Section 1 are normative, except where explicitly noted as non-normative.

The following typographical conventions are used throughout the document:

- ALL CAPS type is used for the special terms from [ISODir2] enumerated above.
- `Monospace` type is used to denote programming language, UML, and XML identifiers, as well as for the text of XML documents.

3 Abstract Data Content

This section specifies the abstract data content of a data file conforming to this standard.

A data file conforming to this standard SHALL consist of:

- An optional LEIFileHeader, as specified in Section 3.1.
- Zero or more LEI Data Records, as specified in Section 3.3.

3.1 LEI File Header

The LEI File Header describes the context for the LEI Data Records contained in the main body of the file. The header exists to answer such questions as where the data came from, when it was collected into this file, etc. The content of the header SHALL NOT be required to interpret the data content of any LEI Data Record; each LEI Data Record is self contained.

The LEI File Header is optional. When included in an LEI Data File, the LEI File Header SHALL include the following data elements:

Element Name	Type	Card	Description
ContentDate	DateTime	0,1	The date and time as of which the data contained in the file is valid. Often, this is the date and time when the data was extracted from a database.

Element Name	Type	Card	Description
Originator	LEI	0..1	The entity that created the content of this file. If the originating entity is a pre-LOU, this value SHALL be the LEI of the pre-LOU as would be used in the ManagingLOU field of an LEI Data Record (Section 3.3.2). If the originating entity is the COU, this value SHALL be the LEI of the GLEIF.
FileContent	FileContentEnum	0..1	A code that describes the content of records
DeltaStart	DateTime	0..1	If FileContent is LOU_DELTA_PUBLISHED or COU_DELTA_PUBLISHED, the date and time of the baseline relative to which this file contains new or changed LEI Data records.
RecordCount	Integer	0..1	The number of LEI Data Records in the file.
Extension	Extension (Section 3.3.3)	0,1	An optional element for including header data beyond the standard data elements in an LEI data file. This may include data specific to an LOU, data specific to a publisher of LEI data, and so on.

180 **3.2 FileContentEnum Code List**

181 A value of type FileContentEnum in a file conforming to this standard SHALL be one of the
182 code strings specified in the following table:

Code	Definition
LOU_FULL_PUBLISHED	The file contains all LEI Data Records published by an LOU (all LEI Data Records for which the LOU is the ManagingLOU as defined in Section 3.3.2) as of the date/time the file is created.
LOU_DELTA_PUBLISHED	The file contains those LEI Data Records published by an LOU (all LEI Data Records for which the LOU is the ManagingLOU as defined in Section 3.3.2) which are new or changed since the DeltaDate specified in the header, as of the date/time the file is created.
COU_FULL_PUBLISHED	The file contains all LEI Data Records published by the COU (including all LEI Data Records from all LOUs) as of the date/time the file is created.

Code	Definition
COU_DELTA_PUBLISHED	The file contains those LEI Data Records published by the COU (including all LEI Data Records from all LOUs) which are new or changed since the DeltaDate specified in the header, as of the date/time the file is created.
QUERY_RESPONSE	The file contains records matching criteria specified in a query. (The mechanism for making queries is outside the scope of this standard.)

183 **3.3 LEI Data Record**

184 An LEI Data Record describes a single LEI. Each LEI Data record in a file conforming to this
 185 standard SHALL include data elements as specified below.

Element Name	Type	Card	Description
LEI	LEI	0,1	The 20-character LEI of the legal entity described by this LEI Data Record. An LEI Data Record SHALL include this field if part of a file of type COU_FULL_PUBLISHED, COU_DELTA_PUBLISHED, LOU_FULL_PUBLISHED, or LOU_DELTA_PUBLISHED (Section 3.1). This field MAY be omitted when a file is used to communicate LEI data between LOUs; for example, to communicate a pending registration for which no LEI is yet assigned.
Entity	Entity (Section 3.3.1)	1	Attributes describing the entity itself
Registration	Registration (Section 3.3.2)	1	Attributes describing the registration of this LEI with an LOU
Extension	Extension (Section 3.3.3)	0,1	An optional element for including data beyond the standard data elements in an LEI data file. This may include data specific to an LOU, data specific to a publisher of LEI data, and so on.

186

187 **3.3.1 Entity Section of LEI Data Record**

188 The Entity section of an LEI Data Record in a file conforming to this standard SHALL
 189 include data elements as specified below.

Element Name	Type	Card	Description
LegalName	Name	1	The Legal Name of the Entity. If an Entity is in a jurisdiction with more than one Legal Name (e.g., in different languages), this is the Primary Legal Name (see otherEntityNames for other names).
OtherEntityNames	Other EntityName	0..n	An optional list of other Name instances for the Entity.
LegalAddress	Address	1	The address of the Entity as recorded in the registration of the Entity in its legal jurisdiction
HeadquartersAddress	Address	1	The address of the headquarters of the Entity
OtherAddresses	OtherAddress	0..n	An optional list of other Address instances for the Entity. This may be used to provide alternative language forms of legal address or headquarters address..
BusinessRegister EntityID	Business Register EntityID	0,1	The identifier of the legal entity as maintained by a business registry in the jurisdiction of legal registration or other registration authority. The entity identification code must be published when the relevant registry is included in the list of sources maintained by the ROC Secretariat as having no IP restrictions on redistribution. (<i>available here</i>). The ROC will publish a list of business registries, and will provide further guidance to pre-LOUs and market participants.
LegalJurisdiction	Jurisdiction Code	0,1	The jurisdiction of legal formation and registration of the entity (and on which the legalForm data element is also dependent).

Element Name	Type	Card	Description
LegalForm	Name	0,1	<p>The legal form of the entity, specified as a language-specific freeform name.</p> <p><i>Explanation (non-normative): The definition of types of legal form is expected to become a normative enumeration (code list) in the future, but the work to define such a list is thought to be beyond the scope of this time-constrained process to agree upon Version 1.0 of this interim standard.</i></p>
AssociatedEntity	Associated	*	<p>Another entity associated with this entity if needed to fully identify this entity or to place it in an appropriate context.</p> <p><i>This data element should only be used to specify an additional Name or LEI value as needed to identify "umbrella funds". The AssociatedEntity element should contain the Name or LEI of the Fund Family to which the umbrella fund designated by the LEI in this record belongs.</i></p> <p><i>'Further discussion of "umbrella funds" can be found in the document 'Reference information on " umbrella funds" for the LEI' available here</i></p> <p><i>Further guidance on fund identification will be provided by the ROC.</i></p>
EntityStatus	EntityStatus Enum	1	<p>The status of the legal entity. This is not to be confused with the status of the registration; see RegistrationStatus.</p> <p>If this LEI record contains a non-empty SuccessorEntity field, EntityStatus is the last status of the legal entity before this record was superseded; this status is not necessarily the current status of the legal entity.</p>
EntityExpirationDate	DateTime	0,1	<p>The date that the legal entity ceased to operate, whether due to dissolution, merger or acquisition. Omitted if the legal entity has not ceased to operate.</p>

Element Name	Type	Card	Description
EntityExpirationReason	EntityExpirationReasonEnum	0,1	The reason that a legal entity ceased to operate. This element SHALL be present if EntityExpirationDate is present, and omitted otherwise.
SuccessorEntity	Successor	0,1	<p>The LEI of the LEI registration that supersedes or subsumes this LEI registration for the same legal entity.</p> <p>If RegistrationStatus is DUPLICATE, then SuccessorEntity is the LEI of the surviving LEI Registration. In this case, the successor SHALL be indicated using an LEI.</p> <p>If RegistrationStatus is MERGED, then SuccessorEntity SHALL be the LEI Registration of the new/acquiring entity when the successor has an LEI. In cases where the successor does not have an LEI, the name form of SuccessorEntity may be used.</p> <p>Otherwise, SuccessorEntity is omitted.</p> <p>When a successor is assigned to an LEI registration, the original LEI registration may no longer be updated (since updates would be applied to the successor registration, if any).</p> <p>As a consequence, other fields of the superseded registration record (address, entity status, etc) may no longer reflect the current state of the legal entity.</p>

190 **3.3.2 Registration Section of LEI Data Record**

191 The Registration section of an LEI Data Record in a file conforming to this standard
192 SHALL include data elements as specified below.

Element Name	Type	Card	Description
InitialRegistrationDate	DateTime	1	Date/time the LEI record was created
LastUpdateDate	DateTime	1	Date/time the LEI record was most recently updated

Element Name	Type	Card	Description
RegistrationStatus	Registration StatusEnum	1	Status of the LEI registration. This is not to be confused with the status of the legal entity itself; see EntityStatus.
NextRenewalDate	DateTime	1	The next date by which the LEI registration should be renewed and re-certified by the legal entity.
ManagingLOU	LEI	1	The LEI of the LOU that is responsible for administering this LEI registration.
ValidationSources	Validation SourcesEnum	0,1	The current validation status of this LEI record, or omitted if the validation status is not known or not revealed.

193

194 3.3.3 Extension Section of LEI Data record

195 The `Extension` section of an LEI record may be used to include additional data not defined in
 196 this standard. For example, an LOU may use `Extension` to publish additional data elements it
 197 collects as part of registration.

198 The following rules must be observed:

- 199 • Each XML element included in the content of the `Extension` element SHALL be in an
 200 XML namespace that is not null and not equal to the XML namespace of the LEI Data File as
 201 specified in this standard.
- 202 • The XML namespace for an extension element SHALL be a namespace to which the creator
 203 of the extension element is entitled to use; e.g., a namespace derived from the Internet
 204 Domain Name of the creator, a namespace agreed upon by a group of trading partners, etc.
- 205 • An extension element SHALL NOT be defined in such a way as to require the recipient of
 206 the file to recognize the extension element in order to interpret the data elements specified in
 207 this standard. A recipient of the file must be able to ignore all extension elements and still
 208 interpret the standard content correctly.
- 209 • A recipient of a data file conforming to this standard SHALL NOT reject a file solely
 210 because it contains extensions not understood by the recipient. A recipient must be prepared
 211 to accept a file containing extensions and ignore any it does not understand, provided that the
 212 file complies to this standard.

213 **3.4 Data Types**

214 This section specifies the data types referenced by the tables in Section 3.3, in alphabetical order.

215

216 **3.4.1 Address Data Type**

217 A value of type `Address` in a file conforming to this standard SHALL include data elements as
218 specified below.

Element Name	Type	Cardinality	Description
<code>xml:lang</code>	<code>LanguageCode</code>	0,1	The language in which all of the string-valued components of this address are expressed.
<code>Line1</code>	<code>String</code>	1	The first line of the street address
<code>Line2</code>	<code>String</code>	0,1	The second line of the street address
<code>Line3</code>	<code>String</code>	0,1	The third line of the street address. This element SHALL be omitted if <code>addressLine2</code> is omitted.
<code>Line4</code>	<code>String</code>	0,1	The fourth line of the street address. This element SHALL be omitted if <code>addressLine3</code> is omitted.
<code>City</code>	<code>String</code>	1	The name of the city
<code>Region</code>	<code>RegionCode</code>	0,1	The code for a region (state, province, county, canton, etc) as used in postal addresses for the country of this address.
<code>Country</code>	<code>CountryCode</code>	1	The 2-character ISO 3166-1 country code of the country
<code>PostalCode</code>	<code>String</code>	0,1	The postal code of this address as specified by the local postal service.

219

220 **3.4.2 AssociatedEntity Data Type**

221 A value of type `AssociatedEntity` in a file conforming to this standard SHALL include
222 data elements as specified below.

Element Name	Type	Card	Description
type	AssociatedEntity	1	The type of association represented by this AssociatedEntity instance. NOTE There is only one legal value for this enumeration – FUND FAMILY and that use is reserved to dealing with “umbrella” fund identification only.
Associated LEI	LEI	0,1	The LEI of the associated entity. Exactly one of AssociatedLEI or AssociatedEntityName SHALL be specified. AssociatedLEI SHOULD be used if the LEI of the associated entity is known.
Associated EntityName	Name	0,1	The name of the associated entity. Exactly one of AssociatedLEI or AssociatedEntityName SHALL be specified. AssociatedLEI SHOULD be used if the LEI of the associated entity is known.

223

224 **3.4.3 BusinessRegisterEntityID Data Type**

225 A BusinessRegisterEntityID specifies identifier of the legal entity as maintained by a
 226 business registry in the jurisdiction of legal registration, or if the entity is one that is not recorded
 227 in a business registry (e.g., one of the varieties of funds registered instead with financial
 228 regulators), the identifier of the entity in the appropriate registration authority.

229 A value of type BusinessRegisterEntityID in a file conforming to this standard SHALL
 230 include data elements as specified below.

Element Name	Type	Card	Description
register	BusinessRegisterEnum	1	A code that identifies the business register or other registration authority that supplied the value of EntityID. It may also be included without EntityID, to indicate that the entity is registered in the specified business register but that the identifier within that register is not available for publication
EntityID	String	0,1	The identifier of the legal entity as maintained by a business registry or other registration authority. If omitted, it indicates that the entity is registered in the specified business register but that the identifier within that register is not available for publication.

231 **3.4.4 CountryCode Data Type**

232 A value of type CountryCode in a file conforming to this standard SHALL be a 2-character
 233 country code conforming to ISO ISO 3166-1 alpha-2 [ISO3166]. Note that ISO 3166-1 alpha-2
 234 codes are all uppercase.

235 **3.4.5 DateTime Data Type**

236 A value of type DateTime in a conforming to this standard SHALL be a point in time
 237 expressed as a string conforming to ISO 8601 having the following format:

238 *YYYY-MM-DDThh:mm:ss.sssTZ*

239 where the components of the above string are as follows:

- 240 • *YYYY* is the year
- 241 • *MM* is the month (01 = January, ..., 12 = December)
- 242 • *DD* is the day of the month (01 = first day of the month)
- 243 • *T* is the single character ‘T’
- 244 • *hh* is the hour (00 – 23)
- 245 • *mm* is the minute

- 246 • *ss.sss* is the second and milliseconds. From one to three digits may be used for
247 milliseconds, or omitted entirely along with the decimal point.
- 248 • *TZ* is the time zone specifier, which can be either:
- 249 • *Z* the single character ‘Z’, denoting Coordinated Universal Time (UTC); or
- 250 • *+hh:mm* denoting a positive offset from UTC; or
- 251 • *-hh:mm* denoting a negative offset from UTC

252 In the XML representation specified in Section 5, the XSD type `xs:dateTime` is used;
253 however, whereas `xs:dateTime` permits the time zone specifier to be omitted, `DateTime`
254 values in files conforming to this standard SHALL always include a time zone specifier.

255 *Explanation (non-normative): milliseconds are hardly necessary for LEI reference data, and*
256 *likewise it might seem simpler to allow only “Z” as a time zone specifier; however, XML*
257 *processing tools support the full syntax given above and it is not always possible to restrict such*
258 *tools to avoid milliseconds or force the use of “Z” as the time zone specifier. The restriction that*
259 *the time zone specifier must be present is equivalent to using XSD type `xs:timestamp`;*
260 *however this was introduced in XSD 1.1 and not supported by the majority of XML processing*
261 *tools which still only implement XSD 1.0.*

262 **3.4.6 JurisdictionCode Data Type**

263 A value of type `JurisdictionCode` in a file conforming to this standard SHALL be a 2-
264 character country code conforming to ISO 3166-1 alpha-2 [ISO3166] or a region code
265 conforming to ISO 3166-2 [ISO3166-2]. Note that ISO 3166-1 alpha-2 codes and ISO 3166-2
266 are all uppercase. An ISO 3166-1 alpha-2 code SHALL be used to indicate a country
267 jurisdiction, and an ISO 3166-2 code SHALL be used to indicate a regional jurisdiction.

268 **3.4.7 LanguageCode Data Type**

269 A value of type `LanguageCode` in a file conforming to this standard SHALL be an IETF
270 Language Code conforming to [RFC4646]. Note that the first characters of an IETF Language
271 Code, up to the hyphen (if any), are all lowercase, and those following the hyphen (if any) are all
272 uppercase.

273 **3.4.8 LEI Data Type**

274 A value of type `LEI` in a file conforming to this standard SHALL be a 20-character Legal Entity
275 Identifier conforming to [ISO17422]. Conformance to [ISO17442] includes having correct
276 check digits.

277 **3.4.9 Name Data Type**

278 A Name is a string expressed in a natural language, including a code indicating which natural
279 language is used.

280 A value of type Name in a file conforming to this standard SHALL include data elements as
281 specified below.

Element Name	Type	Card	Description
xml:lang	LanguageCode	0,1	The language of name
Name	String	1	The name itself.

282 **3.4.10 OtherAddress Data Type**

283 A value of type OtherAddress in a file conforming to this standard SHALL include data
284 elements as specified below. Each Address element includes an optional language code,
285 permitting OtherAddress to be repeated as many times as necessary to express the same
286 address type in multiple languages. The purpose of the AddressTypeEnum code list is to
287 accommodate legal address and headquarters address in different languages, not to add other
288 address types (which could conceivably be added in the future).

Element Name	Type	Card	Description
type	AddressTypeEnum	1	The type of address represented by this OtherAddress instance.
Address	Address	1	The address

289

290 **3.4.11 OtherEntityName Data Type**

291 A value of type OtherEntityName in a file conforming to this standard SHALL include data
292 elements as specified below. Each Name element includes an optional language code, permitting
293 OtherEntityName to be repeated as many times as necessary to express the same name type
294 in multiple languages. When type is PREFERRED_ASCII_TRANSLITERATED_LEGAL or
295 AUTO_ASCII_TRANSLITERATED_LEGAL, the language code specifies the language of the
296 name prior to transliteration.

Element Name	Type	Card	Description
type	EntityNameTypeEnum	1	The type of name represented by this OtherEntityName instance.

Element Name	Type	Card	Description
Name	Name	1	The name. If type is PREFERRED_ASCII_TRANSLITERATED_LEGAL or AUTO_ASCII_TRANSLITERATED_LEGAL, then this value SHALL only include characters from the character set specified in Section 9.

297

298 **3.4.12 RegionCode Data Type**

299 A value of type RegionCode in a file conforming to this standard SHALL be a code
300 conforming to ISO 3166-2. Note that ISO 3166-2 codes are all uppercase.

301 **3.4.13 SuccessorEntity Data Type**

302 A value of type SuccessorEntity in a file conforming to this standard SHALL include data
303 elements as specified below.

Element Name	Type	Card	Description
SuccessorLEI	LEI	0,1	The LEI of the successor entity. Exactly one of SuccessorLEI or SuccessorEntityName SHALL be specified. SuccessorLEI SHALL be used if the LEI of the successor entity is known.
Successor	Name	0,1	The name of the successor entity. Exactly one of SuccessorLEI or SuccessorEntityName SHALL be specified. SuccessorLEI SHALL be used if the LEI of the successor entity is known.

304

305 **3.5 Enumerated Code Lists**

306 This section specifies the enumerated code list data types (all having the suffix Enum) referenced
307 by the tables in Sections 3.3 and 3.3.2, in alphabetical order.

308 **3.5.1 AddressTypeEnum Code List**

309 The AddressTypeEnum value in an OtherAddress instance specifies how the alternative
310 address relates to the legal entity.

311 A value of type AddressTypeEnum in a file conforming to this standard SHALL be one of the code
312 strings specified in the following table:

Code	Definition
LEGAL_ADDRESS	Registered address of the entity in the legal jurisdiction
HEADQUARTERS_ADDRESS	Address of the headquarters of the entity

313

314 **3.5.2 AssociatedEntityTypeEnum Code List**

315 The AssociatedEntityTypeEnum value in a AssociatedEntity specifies how the
316 associated entity relates to the legal entity.

317 A value of type AssociatedEntityTypeEnum in a file conforming to this standard SHALL
318 be one of the code strings specified in the following table:

Code	Definition
FUND_FAMILY	The legal entity is a fund, and the associated entity is the manager of the fund.

319 **3.5.3 BusinessRegisterEnum Code List**

320 The BusinessRegisterEnum value in an Entity instance specifies what business register
321 provided the value of BusinessRegisterEntityID for the legal entity.

322 The value of BusinessRegisterEnum SHALL be a code provided on the list of business
323 register codes as published by the LEI ROC Secretariat.

324 **3.5.4 EntityExpirationReasonEnum Code List**

325 The EntityExpirationReasonEnum value in an LEI record specifies the reason that the
326 legal entity expired.

327 A value of type EntityExpirationReasonEnum in a file conforming to this standard
328 SHALL be one of the code strings specified in the following table:

Code	Definition
DISSOLVED	The entity ceased to operate
CORPORATE_ACTION	The entity was acquired or merged with another entity
OTHER	The reason for expiry is not one of the above

329

330 **3.5.5 EntityNameTypeEnum Code List**

331 The EntityNameTypeEnum value in an EntityName specifies how the name relates to the
332 legal entity.

333 A value of type EntityNameTypeEnum in a file conforming to this standard SHALL be one
334 of the code strings specified in the following table:

Code	Definition
OTHER_LEGAL	Registered name of the entity in an alternate language in the legal jurisdiction in which the entity is registered
PREFERRED_ASCII_TRANSLITERATED_LEGAL	Legal name of the entity transliterated to ASCII characters, provided by the entity for this purpose
AUTO_ASCII_TRANSLITERATED_LEGAL	Legal name of the entity transliterated to ASCII characters, auto-transliterated by the managing LOU

335 **3.5.6 EntityStatusEnum Code List**

336 The EntityStatusEnum value in an LEI record indicates the status of the legal entity itself.
337 This is not to be confused with the status of the LEI registration, which is specified by
338 RegistrationStatusEnum (Section 3.5.7). See also Section 10, which illustrates how the
339 EntityStatusEnum value changes over the lifecycle of an LEI registration.

340 A value of type EntityStatusEnum in a file conforming to this standard SHALL be one of
341 the code strings specified in the following table:

Code	Definition
ACTIVE	As of the last report or update, the legal entity reported that is was legally registered and operating.
INACTIVE	It has been determined that the entity that was assigned the LEI is no longer legally registered and/or operating, whether as a result of: <ol style="list-style-type: none">1. Business closure2. Acquisition by or merger with another (or new) entity3. Determination of illegitimacy

342 **3.5.7 RegistrationStatusEnum Code List**

343 The RegistrationStatusEnum value in an LEI record indicates the status of the
344 registration of the legal entity with an LOU. This is not to be confused with the status of the
345 legal entity itself, which is specified by EntityStateEnum (Section 3.5.6). See also
346 Section 10, which illustrates how the EntityStateEnum value changes over the lifecycle of
347 an LEI registration, and Section 11, which provides further explanation.

348 A value of type RegistrationStatusEnum in a file conforming to this standard SHALL be
349 one of the code strings specified in the following table:

Code	Definition
PENDING_VALIDATION	<p>An application for an LEI that has been submitted and which is being processed and validated.</p> <p><i>NOTE: LEI registrations in the PENDING state are not intended for public release, but could be used internally between LOUs.</i></p>
ISSUED	<p>An LEI Registration that has been validated and issued, and which identifies an entity that was an operating legal entity as of the last update.</p>
DUPLICATE	<p>An LEI Registration that has been determined to be a duplicate registration of the same legal entity as another LEI Registration; the DUPLICATE status is assigned to the non-surviving registration (i.e., the LEI that should no longer be used). Only one of the potential multiple identifiers will survive; for all other duplicate registrations:</p> <ol style="list-style-type: none"><li data-bbox="565 1184 1317 1220">1. The RegistrationStatus is set to DUPLICATE,<li data-bbox="565 1236 1325 1346">2. The LEI of the surviving LEI Registration is set in the SuccessorLEI data element of (each) duplicate LEI registration;<li data-bbox="565 1367 1338 1440">3. The LastUpdateDate is set to reflect the date of this update, and<li data-bbox="565 1459 1422 1528">4. No further updates of the DUPLICATE registration record will occur.

Code	Definition
LAPSED	<p>An LEI registration that has not been renewed and has exceeded any allowed grace period for renewal.</p> <p>After being issued an LEI, an entity must regularly do the following:</p> <ul style="list-style-type: none"> • Periodically verify the continued accuracy of its registration reference data that is recorded in the LOU that is responsible for managing the LEI registration of the entity, updating any aspect of the registration reference data that has changed; • Periodically renew its LEI registration agreement with the LOU, paying the renewal fee. <p>(Although both of the above actions are typically performed at the same time, it is certainly possible that the frequency of each action could be different.)</p> <p>If, after being issued an LEI,</p> <ul style="list-style-type: none"> • A legal entity fails to renew and re-certify its LEI registration with the LOU responsible to manage the registration by the <code>leiNextRenewalDate</code>, and • The legal entity fails to do so for a pre-determined (as yet unspecified) period of time, and • The legal entity is not known by public sources to have ceased operation <p>Then</p> <ol style="list-style-type: none"> 1. The <code>leiRegistrationStatus</code> is set to LAPSED, 2. Updates to the LEI registration are permitted, notably to reinstate the registration to the ISSUED status.

Code	Definition
MERGED	<p>An LEI registration for an entity that has been merged into another legal entity, such that this legal entity no longer exists as an operating entity.</p> <p>If</p> <ul style="list-style-type: none"> • After being issued an LEI, the entity is acquired by, or merged with, another legal entity; • Per agreements among the parties to the transaction, the LEI of the acquired or merged entity will not be used to identify the surviving entity (or if a new entity is created that is issued a new LEI) <p>Then</p> <ol style="list-style-type: none"> 1. The <code>leiRegistrationStatus</code> is set to “MERGED”, 2. The LEI of the surviving/new legal entity is set in the <code>successorLEI</code> data element of (each) LEI registration that is no longer to be used; 3. The <code>leiRecordLastUpdate</code> is set to reflect the date of this update, 4. The <code>EntityExpirationDate</code> is also set to the date of this update, 5. The <code>EntityExpirationReason</code> is set to “CORPORATE_ACTION”, 6. The <code>EntityStatus</code> is set to “INACTIVE”; and 7. No further updates of the MERGED registration record(s) will occur.

Code	Definition
RETIRED	<p>An LEI registration for an entity that has ceased operation, without having been merged into another entity.</p> <p>If</p> <ul style="list-style-type: none"> • After being issued an LEI, the entity ceases to operate (goes out of business and/or dissolves its legal standing) • The entity informs the LOU of the cessation of business, OR, the managing LOU determines by public sources that the legal entity has been dissolved or ceased to operate (and the LOU seeks to confirm this status through all available channels with the entity) <p>Then</p> <ol style="list-style-type: none"> 1. The <code>leiRegistrationStatus</code> is set to “RETIRED”; 2. The <code>leiRecordLastUpdate</code> is set to reflect the date of this update; 3. The <code>EntityExpirationDate</code> is also set to the date of this update; 4. The <code>EntityExpirationReason</code> is set to “DISSOLVED”; 5. The <code>EntityStatus</code> is set to “INACTIVE”; and 6. No further updates of the RETIRED registration record will occur.
ANNULLED	An LEI registration that was marked as erroneous or invalid after it was issued.
CANCELLED	An LEI registration that was abandoned prior to issuance of an LEI. A record in this state is not published, but may be exchanged inter-LOU.
TRANSFERRED	An LEI registration that has been transferred to a different LOU as the managing LOU. A record in this state is not published, but may be used internally by the prior LOU for audit trail purposes.
PENDING_TRANSFER	An LEI registration that has been requested to be transferred to another LOU. The request is being processed at the sending LOU. When the receiving LOU is ready, the status will be changed to <code>PENDING_ARCHIVAL</code> by the sending LOU prior to completion of the transfer.

Code	Definition
PENDING_ARCHIVAL	An LEI registration is about to be transferred to a different LOU, after which its registration status will revert to a non-pending status. The PENDING_ARCHIVAL status serves to inform recipients of LOU-provided data files that an LEI record will be removed from that LOU's published file after the transfer is complete.

350 3.5.8 ValidationSourcesEnum Code List

351 A value of type ValidationSourcesEnum in a file conforming to this standard SHALL be
 352 one of the code strings specified in the following table:

Code	Definition
PENDING	The validation of the reference data provided by the registrant has not yet occurred.
ENTITY_SUPPLIED_ONLY	Based on the validation procedures in use by the LOU responsible for the record, the information associated with this record has significant reliance on the information that a submitter provided due to the unavailability of corroborating information.
PARTIALLY_CORROBORATED	Based on the validation procedures in use by the pre-LOU responsible for the record, the information supplied by the registrant can be partially corroborated by public authoritative sources, while some of the record is dependent upon the information that the registrant submitted, either due to conflicts with authoritative information, or due to data unavailability.
FULLY_CORROBORATED	Based on the validation procedures in use by the pre-LOU responsible for the record, there is sufficient information contained in authoritative public sources to corroborate the information that the submitter has provided for the record.

353 4 Constraints and Data Validation

354 All values of type String specified in Section 3 SHALL be 500 or fewer characters in length,
 355 shall not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters, shall not
 356 begin or end with a space (#x20) character, and shall not include a sequence of two or more
 357 adjacent space characters. Otherwise, all characters in the Universal Character Set specified in
 358 [ISO10646] which match the Char production of [XML1.0] are permitted. (This constraint is
 359 equivalent to saying that the value must fall within the lexical space of xsd:token as defined
 360 in [XSD1.0], limited to 500 or fewer characters.)

361 5 XML Syntax

362 This section specifies the XML schema for an LEI data file conforming to this standard.

363 5.1 XML Design Rules

- 364 • The XSD schema conforms to [XSD1,XSD2]
- 365 • The XML namespace is `http://www.leiroc.org/data/schema/leidata/2014`
- 366 • All interior XML elements are namespace-qualified (element form = qualified)
- 367 • All XML attributes are in the null namespace (attribute form = unqualified), with the
368 exception of `xml:lang`.
- 369 • Element names are upper camel case
- 370 • Attribute name are lower camel case
- 371 • XSD type names are upper camel case
- 372 • Enumeration code list values are all caps with underscores
- 373 • Elements are used in preference to attributes *except* for language and type qualifiers
- 374 • For a data element specified in Section 3 as having unbounded cardinality, the XML includes
375 a single container element whose subelements are one or more instances of the data element
376 whose cardinality is unbounded. The name of the container element is formed as the plural
377 of the name of the contained elements.

378 5.2 XML Schema

379 An XML file conforming to this standard SHALL be valid according to the following XSD 1.0
380 schema.

```
381 <?xml version="1.0" encoding="UTF-8"?>
382 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
383           elementFormDefault="qualified"
384           targetNamespace="http://www.leiroc.org/data/schema/leidata/2014"
385           xmlns:lei="http://www.leiroc.org/data/schema/leidata/2014">
386
387   <xs:import namespace="http://www.w3.org/XML/1998/namespace"
388             schemaLocation="http://www.w3.org/2001/xml.xsd"/>
389
390   <xs:element name="LEIData" type="lei:LEIData" />
391
392   <xs:complexType name="LEIData">
393     <xs:sequence>
394       <xs:element name="LEIHeader" type="lei:LEIHeaderType" minOccurs="0"/>
395       <xs:element name="LEIRecords" type="lei:LEIRecordsType"/>
396     </xs:sequence>
397   </xs:complexType>
398
399   <xs:complexType name="LEIHeaderType">
400     <xs:sequence>
401       <xs:element name="ContentDate" type="xs:dateTime" minOccurs="0"/>
402       <xs:element name="Originator" type="lei:LEIType" minOccurs="0"/>
403       <xs:element name="FileContent" type="lei:FileContentEnum" minOccurs="0"/>
404       <xs:element name="DeltaStart" type="xs:dateTime" minOccurs="0"/>

```

```

405     <xs:element name="RecordCount" type="xs:int" minOccurs="0"/>
406     <xs:element name="NextVersion" type="lei:HeaderNextVersionType" minOccurs="0"/>
407     <xs:element name="Extension" type="lei:ExtensionType" minOccurs="0"/>
408   </xs:sequence>
409 </xs:complexType>
410
411 <xs:complexType name="HeaderNextVersionType">
412   <xs:sequence>
413     <xs:any minOccurs="0" maxOccurs="unbounded" processContents="lax"
414 namespace="##targetNamespace"/>
415   </xs:sequence>
416 </xs:complexType>
417
418 <xs:complexType name="LEIRecordsType">
419   <xs:sequence>
420     <xs:element name="LEIRecord" type="lei:LEIRecordType" minOccurs="0" maxOccurs="unbounded"/>
421   </xs:sequence>
422 </xs:complexType>
423
424 <xs:complexType name="LEIRecordType" >
425   <xs:sequence>
426     <xs:element name="LEI" type="lei:LEIType"/>
427     <xs:element name="Entity" type="lei:EntityType"/>
428     <xs:element name="Registration" type="lei:RegistrationType"/>
429     <xs:element name="NextVersion" type="lei:LEIRecordNextVersionType" minOccurs="0"/>
430     <xs:element name="Extension" type="lei:ExtensionType" minOccurs="0"/>
431   </xs:sequence>
432 </xs:complexType>
433
434 <xs:complexType name="LEIRecordNextVersionType">
435   <xs:sequence>
436     <xs:any minOccurs="0" maxOccurs="unbounded" processContents="lax"
437 namespace="##targetNamespace"/>
438   </xs:sequence>
439 </xs:complexType>
440
441 <xs:complexType name="EntityType" >
442   <xs:sequence>
443     <xs:element name="LegalName" type="lei:NameType"/>
444     <xs:element name="OtherEntityNames" type="lei:OtherEntityNamesType" minOccurs="0"/>
445     <xs:element name="LegalAddress" type="lei:AddressType"/>
446     <xs:element name="HeadquartersAddress" type="lei:AddressType"/>
447     <xs:element name="OtherAddresses" type="lei:OtherAddressesType" minOccurs="0"/>
448     <xs:element name="BusinessRegisterEntityID" type="lei:BusinessRegisterEntityIDType"
449 minOccurs="0"/>
450     <xs:element name="LegalJurisdiction" type="lei:JurisdictionCodeType" minOccurs="0"/>
451     <xs:element name="LegalForm" type="lei:NameType" minOccurs="0"/>
452     <xs:element name="AssociatedEntity" type="lei:AssociatedEntityType" minOccurs="0"/>
453     <xs:element name="EntityStatus" type="lei:EntityStatusEnum"/>
454     <xs:element name="EntityExpirationDate" type="xs:dateTime" minOccurs="0"/>
455     <xs:element name="EntityExpirationReason" type="lei:EntityExpirationReasonEnum"
456 minOccurs="0"/>
457     <xs:element name="SuccessorEntity" type="lei:SuccessorEntityType" minOccurs="0"/>
458     <xs:element name="NextVersion" type="lei:EntityNextVersionType" minOccurs="0"/>
459   </xs:sequence>
460 </xs:complexType>
461
462 <xs:complexType name="EntityNextVersionType">
463   <xs:sequence>
464     <xs:any minOccurs="0" maxOccurs="unbounded" processContents="lax"
465 namespace="##targetNamespace"/>
466   </xs:sequence>
467 </xs:complexType>
468
469 <xs:complexType name="RegistrationType" >
470   <xs:sequence>
471     <xs:element name="InitialRegistrationDate" type="xs:dateTime"/>
472     <xs:element name="LastUpdateDate" type="xs:dateTime"/>
473     <xs:element name="RegistrationStatus" type="lei:RegistrationStatusEnum"/>
474     <xs:element name="NextRenewalDate" type="xs:dateTime" />
475     <xs:element name="ManagingLOU" type="lei:LEIType"/>

```

```

476     <xs:element name="ValidationSources" type="lei:ValidationSourcesEnum" minOccurs="0"/>
477     <xs:element name="NextVersion" type="lei:RegistrationNextVersionType" minOccurs="0"/>
478   </xs:sequence>
479 </xs:complexType>
480
481 <xs:complexType name="RegistrationNextVersionType">
482   <xs:sequence>
483     <xs:any minOccurs="0" maxOccurs="unbounded" processContents="lax"
484 namespace="##targetNamespace"/>
485   </xs:sequence>
486 </xs:complexType>
487
488 <xs:complexType name="ExtensionType">
489   <xs:sequence>
490     <xs:any minOccurs="0" maxOccurs="unbounded" processContents="lax" namespace="##other"/>
491   </xs:sequence>
492 </xs:complexType>
493
494 <xs:complexType name="AddressType">
495   <xs:sequence>
496     <xs:element name="Line1" type="lei:Token500Type"/>
497     <xs:element name="Line2" type="lei:Token500Type" minOccurs="0"/>
498     <xs:element name="Line3" type="lei:Token500Type" minOccurs="0"/>
499     <xs:element name="Line4" type="lei:Token500Type" minOccurs="0"/>
500     <xs:element name="City" type="lei:Token500Type" />
501     <xs:element name="Region" type="lei:RegionCodeType" minOccurs="0"/>
502     <xs:element name="Country" type="lei:CountryCodeType" />
503     <xs:element name="PostalCode" type="lei:Token500Type" minOccurs="0"/>
504   </xs:sequence>
505   <xs:attribute ref="xml:lang" use="optional"/>
506 </xs:complexType>
507
508 <xs:complexType name="AssociatedEntityType">
509   <xs:choice>
510     <xs:element name="AssociatedLEI" type="lei:LEIType"/>
511     <xs:element name="AssociatedEntityName" type="lei:NameType"/>
512   </xs:choice>
513   <xs:attribute name="type" type="lei:AssociatedEntityTypeEnum" use="required"/>
514 </xs:complexType>
515
516 <xs:complexType name="BusinessRegisterEntityIDType">
517   <xs:simpleContent>
518     <xs:extension base="lei:Token500Type">
519       <xs:attribute name="register" type="lei:BusinessRegisterEnum"/>
520     </xs:extension>
521   </xs:simpleContent>
522 </xs:complexType>
523
524 <xs:simpleType name="JurisdictionCodeType">
525   <xs:union memberTypes="lei:CountryCodeType lei:RegionCodeType" />
526 </xs:simpleType>
527
528 <xs:complexType name="NameType">
529   <xs:simpleContent>
530     <xs:extension base="lei:Token500Type">
531       <xs:attribute ref="xml:lang" use="optional"/>
532     </xs:extension>
533   </xs:simpleContent>
534 </xs:complexType>
535
536 <xs:complexType name="OtherAddressType">
537   <xs:complexContent>
538     <xs:extension base="lei:AddressType">
539       <xs:attribute name="type" type="lei:AddressTypeEnum" use="required"/>
540     </xs:extension>
541   </xs:complexContent>
542 </xs:complexType>
543
544 <xs:complexType name="OtherAddressesType">
545   <xs:sequence>
546     <xs:element name="OtherAddress" type="lei:OtherAddressType" maxOccurs="unbounded"/>

```

```

547     </xs:sequence>
548 </xs:complexType>
549
550
551 <xs:complexType name="OtherEntityNamesType">
552   <xs:sequence>
553     <xs:element name="OtherEntityName" type="lei:OtherEntityNameType" maxOccurs="unbounded"/>
554   </xs:sequence>
555 </xs:complexType>
556
557 <xs:complexType name="OtherEntityNameType">
558   <xs:complexContent>
559     <xs:extension base="lei:NameType">
560       <xs:attribute name="type" type="lei:EntityTypeEnum" use="required"/>
561     </xs:extension>
562   </xs:complexContent>
563 </xs:complexType>
564
565 <xs:complexType name="SuccessorEntityType">
566   <xs:choice>
567     <xs:element name="SuccessorLEI" type="lei:LEIType"/>
568     <xs:element name="SuccessorEntityName" type="lei:NameType"/>
569   </xs:choice>
570 </xs:complexType>
571
572
573
574 <xs:simpleType name="CountryCodeType">
575   <xs:restriction base="xs:string">
576     <xs:minLength value="2"/>
577     <xs:maxLength value="2"/>
578     <xs:pattern value="[A-Z]{2}"/>
579   </xs:restriction>
580 </xs:simpleType>
581
582 <xs:simpleType name="RegionCodeType">
583   <xs:restriction base="xs:string">
584     <xs:minLength value="4"/>
585     <xs:maxLength value="6"/>
586     <xs:pattern value="[A-Z]{2}-[A-Z0-9]{1,3}"/>
587   </xs:restriction>
588 </xs:simpleType>
589
590 <xs:simpleType name="LEIType">
591   <xs:restriction base="xs:string">
592     <xs:minLength value="20"/>
593     <xs:maxLength value="20"/>
594     <xs:pattern value="[0-9A-Z]{18}[0-9]{2}"/>
595   </xs:restriction>
596 </xs:simpleType>
597
598 <xs:simpleType name="Token500Type">
599   <xs:restriction base="xs:token">
600     <xs:maxLength value="500"/>
601   </xs:restriction>
602 </xs:simpleType>
603
604 <!--
605   All enumeration types provide for forward compatibility by accepting any string as valid at
606   the schema level.
607   The legal values for Version 1.0 are indicated in a 1.0-suffixed type which is unioned with
608   xs:string to
609   form the complete enumeration type.
610   -->
611
612 <xs:simpleType name="FileContentEnum1.0">
613   <xs:restriction base="xs:string">
614     <!-- Enumeration values for 1.0: -->
615     <xs:enumeration value="LOU_FULL_PUBLISHED" />
616     <xs:enumeration value="LOU_DELTA_PUBLISHED" />
617     <xs:enumeration value="COU_FULL_PUBLISHED" />

```

```

618     <xs:enumeration value="COU_DELTA_PUBLISHED" />
619     <xs:enumeration value="QUERY_RESPONSE" />
620   </xs:restriction>
621 </xs:simpleType>
622
623 <xs:simpleType name="FileContentEnum">
624   <xs:union memberTypes="lei:FileContentEnum1.0 xs:string" />
625 </xs:simpleType>
626
627 <xs:simpleType name="BusinessRegisterEnum">
628   <xs:restriction base="xs:string">
629     <xs:annotation>
630       <xs:documentation>
631         <!-- Enumeration values for 1.0 are maintained elsewhere -->
632       </xs:documentation>
633     </xs:annotation>
634   </xs:restriction>
635 </xs:simpleType>
636
637 <xs:simpleType name="EntityNameTypeEnum1.0">
638   <xs:restriction base="xs:string">
639     <!-- Enumeration values for 1.0: -->
640     <xs:enumeration value="OTHER_LEGAL" />
641     <xs:enumeration value="PREFERRED_ASCII_TRANSLITERATED_LEGAL" />
642     <xs:enumeration value="AUTO_ASCII_TRANSLITERATED_LEGAL" />
643   </xs:restriction>
644 </xs:simpleType>
645
646 <xs:simpleType name="EntityNameTypeEnum">
647   <xs:union memberTypes="lei:EntityNameTypeEnum1.0 xs:string" />
648 </xs:simpleType>
649
650 <xs:simpleType name="AddressTypeEnum1.0">
651   <xs:restriction base="xs:string">
652     <!-- Enumeration values for 1.0: -->
653     <xs:enumeration value="LEGAL_ADDRESS" />
654     <xs:enumeration value="HEADQUARTERS_ADDRESS" />
655   </xs:restriction>
656 </xs:simpleType>
657
658 <xs:simpleType name="AddressTypeEnum">
659   <xs:union memberTypes="lei:AddressTypeEnum1.0 xs:string" />
660 </xs:simpleType>
661
662 <xs:simpleType name="AssociatedEntityTypeEnum1.0">
663   <xs:restriction base="xs:string">
664     <!-- Enumeration values for 1.0: -->
665     <xs:enumeration value="FUND_FAMILY" />
666   </xs:restriction>
667 </xs:simpleType>
668
669 <xs:simpleType name="AssociatedEntityTypeEnum">
670   <xs:union memberTypes="lei:AssociatedEntityTypeEnum1.0 xs:string" />
671 </xs:simpleType>
672
673 <xs:simpleType name="EntityStatusEnum1.0">
674   <xs:restriction base="xs:string">
675     <!-- Enumeration values for 1.0: -->
676     <xs:enumeration value="ACTIVE" />
677     <xs:enumeration value="INACTIVE" />
678   </xs:restriction>
679 </xs:simpleType>
680
681 <xs:simpleType name="EntityStatusEnum">
682   <xs:union memberTypes="lei:EntityStatusEnum1.0 xs:string" />
683 </xs:simpleType>
684
685 <xs:simpleType name="EntityExpirationReasonEnum1.0">
686   <xs:restriction base="xs:string">
687     <!-- Enumeration values for 1.0: -->
688     <xs:enumeration value="DISSOLVED" />

```



```

689     <xs:enumeration value="CORPORATE_ACTION" />
690     <xs:enumeration value="OTHER" />
691   </xs:restriction>
692 </xs:simpleType>
693
694 <xs:simpleType name="EntityExpirationReasonEnum">
695   <xs:union memberTypes="lei:EntityExpirationReasonEnum1.0 xs:string" />
696 </xs:simpleType>
697
698 <xs:simpleType name="RegistrationStatusEnum1.0">
699   <xs:restriction base="xs:string">
700     <!-- Enumeration values for 1.0: -->
701     <xs:enumeration value="PENDING_VALIDATION" />
702     <xs:enumeration value="ISSUED" />
703     <xs:enumeration value="DUPLICATE" />
704     <xs:enumeration value="LAPSED" />
705     <xs:enumeration value="MERGED" />
706     <xs:enumeration value="RETIRED" />
707     <xs:enumeration value="ANNULLED" />
708     <xs:enumeration value="CANCELLED" />
709     <xs:enumeration value="TRANSFERRED" />
710     <xs:enumeration value="PENDING_TRANSFER" />
711     <xs:enumeration value="PENDING_ARCHIVAL" />
712   </xs:restriction>
713 </xs:simpleType>
714
715 <xs:simpleType name="RegistrationStatusEnum">
716   <xs:union memberTypes="lei:RegistrationStatusEnum1.0 xs:string" />
717 </xs:simpleType>
718
719 <xs:simpleType name="ValidationSourcesEnum1.0">
720   <xs:restriction base="xs:string">
721     <!-- Enumeration values for 1.0: -->
722     <xs:enumeration value="PENDING" />
723     <xs:enumeration value="ENTITY_SUPPLIED_ONLY" />
724     <xs:enumeration value="PARTIALLY_CORROBORATED" />
725     <xs:enumeration value="FULLY_CORROBORATED" />
726   </xs:restriction>
727 </xs:simpleType>
728
729 <xs:simpleType name="ValidationSourcesEnum">
730   <xs:union memberTypes="lei:ValidationSourcesEnum1.0 xs:string" />
731 </xs:simpleType>
732
733 </xs:schema>
734

```

735 6 Change Management

736 Changes to this standard that affect the data schema SHALL be made by approval and
737 publication of a new version of this document. A new version SHALL be one of the following:

- 738 • *Errata Version* An errata version makes corrections to the normative content of the standard
739 (excluding corrections which would change the data schema) and/or makes changes to non-
740 normative content such as explanatory material. An errata version does not change the XML
741 schema, and so does not affect the interoperability of systems implementing the standard. An
742 errata version is indicated by incrementing the third version number; e.g., 1.0 to 1.0.1, or
743 1.0.1 to 1.0.2.
- 744 • *Minor Version* A minor version may include all changes permitted in an errata version, and
745 in addition adds one or more data elements and/or adds one or more codes to a code list
746 (“enum” data type). A minor version changes the XML schema. Minor version changes to

747 schema follow the procedure in Section 6.1 which provides for forward and backward
748 compatibility. This allows existing implementations to continue to interoperate even if they
749 are using different minor versions. A minor version is indicated by incrementing the second
750 version number; e.g., 1.0 to 1.1 or 1.1.3 to 1.2.

751 • *Major Version* A major version may make any change at all, including incompatible changes
752 to the XML schema. Major version changes to schema follow the procedure in Section 6.2
753 which specifies that the new version uses a different XML namespace. This requires existing
754 implementations to separately understand both the old and new versions during a period of
755 transition. A major version is indicated by incrementing the first version number; e.g., 1.1 to
756 2.0.

757 The release of a new minor or major version shall always be accompanied by a transition plan
758 for LOUs and the COU, to ensure a smooth and time-bounded migration to the new version.

759 **6.1 Minor Version Changes to the XML Schema**

760 A minor version may introduce new XML elements and/or adds one or more codes to a code list
761 (“enum” data type). Minor version changes to schema SHALL be made as specified below, in
762 order to achieve forward and backward compatibility. Forward compatibility means that an LEI
763 Data File which is valid according to the older version’s schema is also valid according to the
764 newer version’s schema. Backward compatibility means that an LEI Data File which is valid
765 according to the newer version’s schema is also valid according to the older version’s schema.

766 New data elements may be added at pre-defined extension points within the schema.
767 Corresponding to each extension point is an optional XML element <NextVersion>. New
768 data elements are always added within a <NextVersion> element. When a minor version
769 adds a new data element to a <NextVersion> element, a new <NextVersion> element is
770 also added inside the old <NextVersion> element, to accommodate additional data elements
771 in subsequent minor versions.

772 To illustrate, here is the <Entity> portion of an LEI Data Record in the first minor version:

```
773 <Entity>  
774   ...  
775   <SuccessorLEI>...</SuccessorLEI>  
776 </Entity>
```

777 If the next minor version adds two data elements to the <Entity> portion, a data file
778 conforming to that minor version looks like this:

```
779 <Entity>  
780   ...  
781   <SuccessorLEI>...</SuccessorLEI>  
782   <NextVersion>  
783     <NewElement1>...</NewElement1>  
784     <NewElement2>...</NewElement2>  
785   </NextVersion>
```

786 </Entity>

787 If the next minor version after that adds one more data elements to the <Entity> portion, a
788 data file conforming to that minor version looks like this:

```
789 <Entity>
790   ...
791   <SuccessorLEI>...</SuccessorLEI>
792   <NextVersion>
793     <NewElement1>...</NewElement1>
794     <NewElement2>...</NewElement2>
795     <NextVersion>
796       <NewElement3>...</NewElement3>
797     </NextVersion>
798   </NextVersion>
799 </Entity>
```

800 The following rules SHALL be observed to ensure forward and backward compatibility:

801 The initial XSD declaration for a <NextVersion> element SHALL be as follows:

```
802 <xsd:element name="NextVersion" type="lei:NextVersion1Type"
803 minOccurs="0"/>
804
805 <xsd:complexType name="NextVersion1Type">
806   <xsd:sequence>
807     <xsd:any minOccurs="0" maxOccurs="unbounded" processContent="lax"
808 namespace="##targetNamespace" />
809   </xsd:sequence>
810 </xsd:complexType>
```

811 The minOccurs declaration on the <NextVersion> element allows it to be omitted in files
812 conforming to the first minor version. The schema wildcard (<xsd:any>) allows for forward
813 compatibility: a file conforming to a new minor version still validates in the old version because
814 the wildcard matches any new elements introduced in the new minor version.

815 New elements SHALL be introduced in a subsequent minor version by modifying the declaration
816 for the above type declaration as follows:

```
817 <xsd:complexType name="NextVersion1Type">
818   <xsd:sequence>
819     <xsd:element name="NewElement1" type="..." minOccurs="0" />
820     <xsd:element name="NewElement2" type="..." minOccurs="0" />
821     <xsd:element name="NextVersion2Type" type="..." minOccurs="0" />
822   </xsd:sequence>
823 </xsd:complexType>
824 <xsd:complexType name="NextVersion2Type">
825   <xsd:sequence>
826     <xsd:any minOccurs="0" maxOccurs="unbounded" processContent="lax"
827 namespace="##targetNamespace" />
828   </xsd:sequence>
829 </xsd:complexType>
```

830 Each new element SHALL be declared minOccurs="0", to ensure backward compatibility: a file
831 conforming to the old version still validates in the new version because the new schema does not
832 require the presence of elements not defined in the old version. If a new element is mandatory
833 for conformance to the new version, this must be enforced outside schema validation.

834 The new definition of the <NextVersion> element SHALL include a declaration of an inner
835 <NextVersion> element, as illustrated above, to provide for additional elements in
836 subsequent minor versions. The nesting of <NextVersion> elements is required to satisfy the
837 "unique particle attribution" constraint of XSD 1.0.

838 Each code list (Enum types) is implemented in the XML schema simply as the XSD string
839 data type. This provides for forward compatibility because the schema for an older minor
840 version will validate any string, including codes defined in newer minor versions. The schema
841 for each minor version includes the list of valid codes for that minor version as a documentation
842 annotation to the type declaration for each Enum type.

843 **6.2 Major Version Changes to the XML Schema**

844 A major version may make any change to the XML schema whatsoever, including incompatible
845 changes.

846 A schema introduced in a new major version SHALL use an XML namespace URI that is
847 different from the XML namespace URI defined in any other major version of this standard. The
848 namespace URI for a new major version SHOULD be the same as the namespace URI specified
849 in Section 5, with the year at the end changed to the year in which the new major version is
850 introduced. If more than one major version is introduced in the same year, a letter "a", "b", "c",
851 etc, may be appended to the year as needed.

852 A new major version must be accompanied by an implementation plan which explains how
853 implementations will make the transition from the old major version to the new major version.
854 Generally speaking, such a plan typically provides for a period of transition in which an
855 implementation capable of receiving the new major version is required to also receive the old
856 major version.

857 **7 Examples (non-normative)**

858 The following is an example of an LEI Data File conforming to XML syntax defined this
859 standard, containing a single LEI record.

```
860 <?xml version="1.0" encoding="UTF-8"?>
861 <lei:LEIData xmlns:lei="http://www.leiroc.org/data/schema/leidata/2014"
862             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
863   <lei:LEIHeader/>
864   <lei:LEIRecords>
865     <lei:LEIRecord>
866       <lei:LEI>12345678901234567890</lei:LEI>
867       <lei:Entity>
868         <lei:LegalName xml:lang="en">Example Corporation</lei:LegalName>
869         <lei:OtherEntityNames>
```

```

870         <lei:OtherEntityName xml:lang="fr" type="OTHER_LEGAL">Exemple
871 Société</lei:OtherEntityName>
872         <lei:OtherEntityName xml:lang="fr" type="AUTO_ROMANIZED_LEGAL">Exemple
873 Societe</lei:OtherEntityName>
874     </lei:OtherEntityNames>
875     <lei:LegalAddress xml:lang="en">
876         <lei:Line1>1234 Legal Entity Lane</lei:Line1>
877         <lei:Line2>Suite 4321</lei:Line2>
878         <lei:City>Exampleville</lei:City>
879         <lei:Region>US-MA</lei:Region>
880         <lei:Country>US</lei:Country>
881         <lei:PostalCode>02400</lei:PostalCode>
882     </lei:LegalAddress>
883     <lei:HeadquartersAddress xml:lang="en">
884         <lei:Line1>One Large Office Building</lei:Line1>
885         <lei:Line2>500th Floor</lei:Line2>
886         <lei:Line3>Suite 444</lei:Line3>
887         <lei:City>Côte d'Exemple</lei:City>
888         <lei:Region>FR-D</lei:Region>
889         <lei:Country>FR</lei:Country>
890     </lei:HeadquartersAddress>
891     <lei:OtherAddresses>
892         <lei:OtherAddress xml:lang="fr" type="HEADQUARTERS_ADDRESS">
893             <lei:Line1>Un Grand Immeuble de Bureaux</lei:Line1>
894             <lei:Line2>500e étage</lei:Line2>
895             <lei:Line3>Suite 444</lei:Line3>
896             <lei:City>Côte d'Exemple</lei:City>
897             <lei:Region>FR-D</lei:Region>
898             <lei:Country>FR</lei:Country>
899             <lei:PostalCode>lei:PostalCode</lei:PostalCode>
900         </lei:OtherAddress>
901     </lei:OtherAddresses>
902     <lei:BusinessRegisterEntityID register="BR-US-MA">2-
903 718281828</lei:BusinessRegisterEntityID>
904     <lei:LegalJurisdiction>US-MA</lei:LegalJurisdiction>
905     <lei:LegalForm xml:lang="en">C Corporation</lei:LegalForm>
906     <lei:AssociatedEntity type="FUND_MANAGER">
907         <lei:AssociatedLEI>98765432109876543210</lei:AssociatedLEI>
908     </lei:AssociatedEntity>
909     <lei:EntityStatus>INACTIVE</lei:EntityStatus>
910     <lei:EntityExpirationDate>2013-12-31T01:02:03Z</lei:EntityExpirationDate>
911     <lei:EntityExpirationReason>CORPORATE_ACTION</lei:EntityExpirationReason>
912     <lei:SuccessorEntity>
913         <lei:SuccessorLEI>1732050807568000000</lei:SuccessorLEI>
914     </lei:SuccessorEntity>
915 </lei:Entity>
916 <lei:Registration>
917     <lei:InitialRegistrationDate>2001-11-30T12:00:00Z</lei:InitialRegistrationDate>
918     <lei:LastUpdateDate>2014-03-15T12:00:00Z</lei:LastUpdateDate>
919     <lei:RegistrationStatus>MERGED</lei:RegistrationStatus>
920     <lei:NextRenewalDate>2015-12-31T01:02:03Z</lei:NextRenewalDate>
921     <lei:ManagingLOU>31415926535897932384</lei:ManagingLOU>
922     <lei:ValidationSources>FULLY_CORROBORATED</lei:ValidationSources>
923 </lei:Registration>
924 </lei:LEIRecord>
925 </lei:LEIRecords>
926 </lei:LEIData>
927

```

928 8 References

929 [ISO646] ISO, "Information technology -- ISO 7-bit coded character set for information
930 interchange," ISO/IEC 646:1991.

931 [ISO3166-1] ISO, "Codes for the representation of names of countries and their subdivisions --
932 Part 1: Country codes," ISO 3166-1:2013.

933 [ISO3166-2] ISO, “Codes for the representation of names of countries and their subdivisions –
934 Part 2: Country subdivision code,” ISO 3166-2:2013.

935 [ISO8601] ISO, “Data elements and interchange formats – Information interchange –
936 Representation of dates and times,” ISO 8601:2004.

937 [ISO10646] ISO, “Information technology -- Universal Coded Character Set (UCS),” ISO
938 10646:2012.

939 [ISO17422] ISO, “Financial Services - Legal Entity Identifier (LEI),” ISO/DIS 17442:2012.

940 [ISODir2] ISO, “Rules for the structure and drafting of International Standards (ISO/IEC
941 Directives, Part 2, 2001, 4th edition),” July 2002.

942 [XSD1] H. Thompson, D. Beech, M. Maloney, N. Mendelsohn, “XML Schema Part 1:
943 Structures,” W3C Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-1/>.

944 [XSD2] P. Biron, A. Malhotra, “XML Schema Part 2: Datatypes,” W3C Recommendation,
945 May 2001, <http://www.w3.org/TR/xmlschema-2/>.

946 **9 Appendix: Character Codes Allowed in ASCII** 947 **Transliterated Names**

948 When a Name instance is of type PREFERRED_ASCII_TRANSLITERATED_LEGAL or
949 AUTO_ASCII_TRANSLITERATED_LEGAL, the value of the name field SHALL consist only
950 of non-control characters drawn from the “invariant subset” of ISO 646. These characters are
951 enumerated below. The “Hex Value” column indicates the code point value (expressed in
952 hexadecimal) for each character in both ISO 646 and ISO 10646.

Graphic Symbol	Name	Hex Value	Graphic Symbol	Name	Hex Value
!	Exclamation Mark	21	M	Capital Letter M	4D
"	Quotation Mark	22	N	Capital Letter N	4E
%	Percent Sign	25	O	Capital Letter O	4F
&	Ampersand	26	P	Capital Letter P	50
'	Apostrophe	27	Q	Capital Letter Q	51
(Left Parenthesis	28	R	Capital Letter R	52
)	Right Parenthesis	29	S	Capital Letter S	53
*	Asterisk	2A	T	Capital Letter T	54
+	Plus sign	2B	U	Capital Letter U	55
,	Comma	2C	V	Capital Letter V	56
-	Hyphen/ Minus	2D	W	Capital Letter W	57

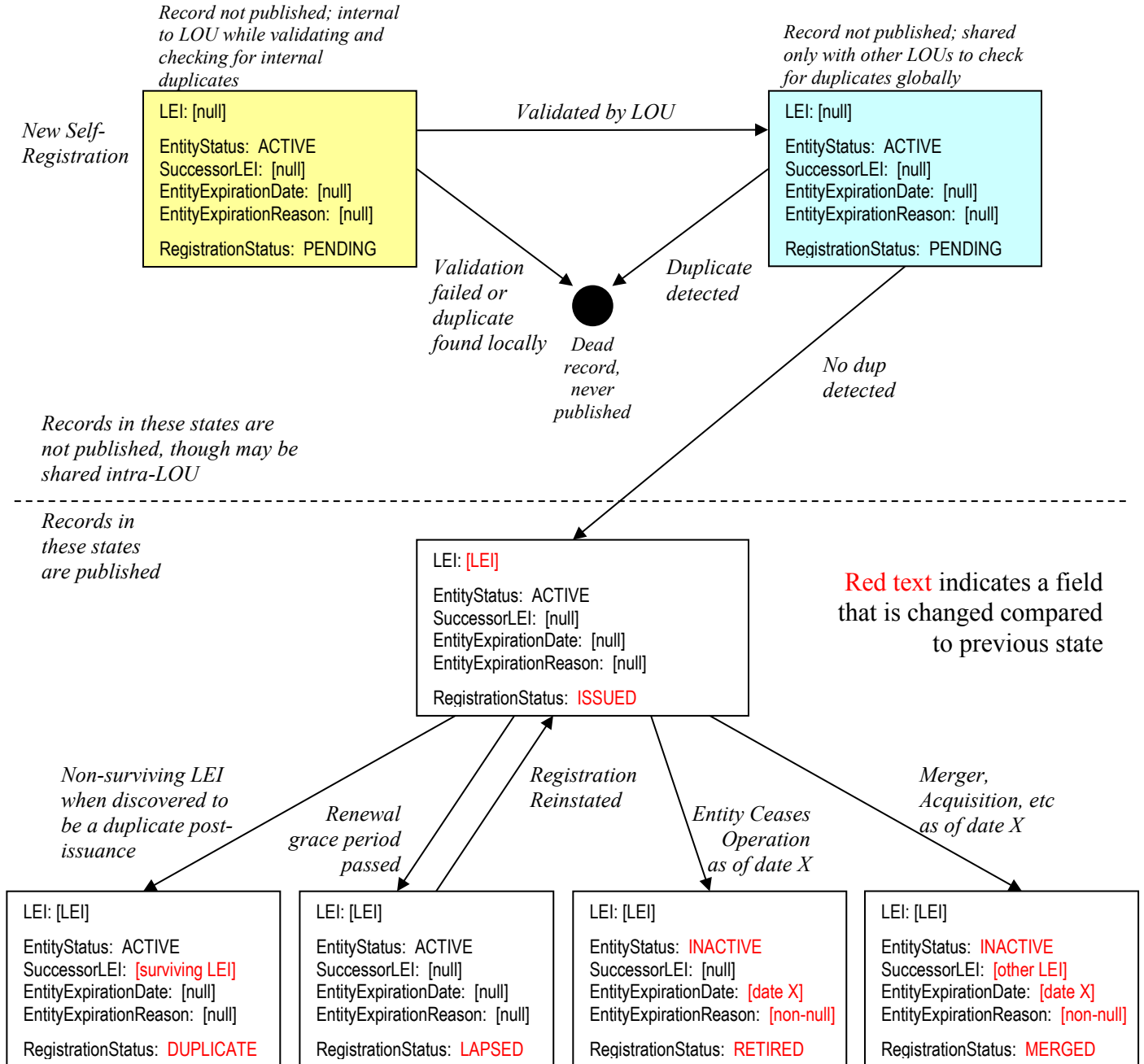
Graphic Symbol	Name	Hex Value	Graphic Symbol	Name	Hex Value
.	Full Stop	2E	X	Capital Letter X	58
/	Solidus	2F	Y	Capital Letter Y	59
0	Digit Zero	30	Z	Capital Letter Z	5A
1	Digit One	31	_	Low Line	5F
2	Digit Two	32	a	Small Letter a	61
3	Digit Three	33	b	Small Letter b	62
4	Digit Four	34	c	Small Letter c	63
5	Digit Five	35	d	Small Letter d	64
6	Digit Six	36	e	Small Letter e	65
7	Digit Seven	37	f	Small Letter f	66
8	Digit Eight	38	g	Small Letter g	67
9	Digit Nine	39	h	Small Letter h	68
:	Colon	3A	i	Small Letter i	69
;	Semicolon	3B	j	Small Letter j	6A
<	Less-than Sign	3C	k	Small Letter k	6B
=	Equals Sign	3D	l	Small Letter l	6C
>	Greater-than Sign	3E	m	Small Letter m	6D
?	Question Mark	3F	n	Small Letter n	6E
A	Capital Letter A	41	o	Small Letter o	6F
B	Capital Letter B	42	p	Small Letter p	70
C	Capital Letter C	43	q	Small Letter q	71
D	Capital Letter D	44	r	Small Letter r	72
E	Capital Letter E	45	s	Small Letter s	73
F	Capital Letter F	46	t	Small Letter t	74
G	Capital Letter G	47	u	Small Letter u	75
H	Capital Letter H	48	v	Small Letter v	76
I	Capital Letter I	49	w	Small Letter w	77
J	Capital Letter J	4A	x	Small Letter x	78
K	Capital Letter K	4B	y	Small Letter y	79
L	Capital Letter L	4C	z	Small Letter z	7A

Graphic Symbol	Name	Hex Value	Graphic Symbol	Name	Hex Value
	Space	20			

953

954 **10 Appendix: LEI Record Transition Diagram**

955 The following diagram illustrates the relationship between status fields of the LEI record.



956

957 In rare circumstances, a record may pass directly from PENDING to RETIRED or MERGED; e.g.,
 958 retroactive reporting of a failed or merged entity. In that case, the EntityStatus would be
 959 INACTIVE while the record is in the PENDING state.

960 Not shown in the diagram are possible transitions from ISSUED to ANNULLED (remaining
 961 below the line), from PENDING to CANCELLED (remaining above the line), and from ISSUED
 962 to TRANSFERRED (the latter being a record that is neither published nor shared intra-LOU, but
 963 maintained only internally by the sending LOU).

964 **11 Appendix: Considerations Regarding RegistrationStatus** 965 **(non-normative)**

966 The RegistrationStatus operational data element indicates the state that an LEI
 967 registration record can have over its life-cycle.

968 The defined status codes fall into three broad categories:

- 969 1. Normal registration life cycle: These are codes applied in the normal life cycle of an LEI
 970 registration from submission through issuance through the termination of the registration if
 971 and when the entity becomes inactive.
- 972 2. Registration Errors: These are codes that are needed to correct errors that may occur in LEI
 973 registration procedures
- 974 3. Transient conditions: These are codes that support certain transient conditions having to do
 975 with transfer of the management of an LEI registration from one LOU to another

976 The following table categorizes the RegistrationStatusEnum codes defined in
 977 Section 3.5.7.

RegistrationStatus Value	Category	Record Publication
PENDING_VALIDATION	Normal life cycle	Internal
ISSUED	Normal life cycle	Published
DUPLICATE	Registration Errors	Published
LAPSED	Normal life cycle	Published
MERGED	Normal life cycle	Published
RETIRED	Normal life cycle	Published
ANNULLED	Registration Errors	Published
CANCELLED	Registration Errors	Archived
TRANSFERRED	Transient condition	Archived
PENDING_TRANSFER	Transient condition	Transient

RegistrationStatus Value	Category	Record Publication
PENDING_ARCHIVAL	Transient condition	Transient

978
 979 Each RegistrationStatus code also implies something about how the LEI registration record is
 980 published, as indicated in the Record Publication column in the table above:

- 981 • *Published* The LEI record is included in the public files published by the managing LOU
 982 and the COU.
- 983 • *Transient* The LEI record is included in the public file published by the managing LOU, but
 984 only for a transient period during a record transfer from one LOU to another.
- 985 • *Archived* The LEI record is not included in the public file published by the managing LOU
 986 or by the CO, but is retained by the managing LOU as part of its archival records
- 987 • *Internal* The LEI record is not included in the public file published by the managing LOU
 988 or by the COU, but may be exchanged between LOUs and/or the COU as part of the
 989 validation process prior to issuance.

990 **11.1 Normal Lifecycle**

991 PENDING_VALIDATION and ISSUED are the most basic registration status codes. If an entity
 992 registers and receives an LEI, and never ceases operation or transfers its registration to another
 993 LOU, these two codes are the only ones ever used.

994 PENDING_VALIDATION is “Internal” with respect to publication because LEI registrations are
 995 expected to be fully validated prior to the LEI being assigned, issued, and published. Should
 996 reference data for LEI registrations whose validations are pending be shared for purposes of
 997 avoiding duplicate LEI registrations prior to issuance, a general principle is that an LEI code
 998 does not accompany this reference data.

999 A normal part of the lifecycle of an LEI registration is the timely update and recertification of its
 1000 reference data. The registration status code LAPSED distinguishes between LEI registrations
 1001 that are “current” (i.e., in good standing, paid up, and updated) and those whose renewals and
 1002 required periodic maintenance updates are “overdue.”

1003 **11.2 Changes in Registration Status When the Status of the Legal**
 1004 **Entity Changes**

1005 Two transitions in the status of a legal entity that will cause a change in the status of the LEI
 1006 registration record for the legal entity are (1) the cessation of business, or closure (legal
 1007 termination) of the legal entity (registration status RETIRED), and (2) a merger or acquisition of
 1008 the legal entity by another legal entity that causes the first legal entity to cease to exist as a
 1009 separate entity (registration status MERGED). These are “end states” of an LEI registration
 1010 record (as shown in the state transition diagram in Section 10), and an update that assigns either

1011 of these two registration status codes to the LEI registration record is the last update that is made
1012 to that record. In order to satisfy the ISO 17442 persistence criteria of the LEI (i.e., the ability to
1013 properly interpret existing and historical records and reports that refer to LEIs in this state), these
1014 LEI registrations records are still part of the “Published” LEI registrations.

1015 **11.3 Changes in Registration Status When a Registration error is** 1016 **Resolved**

1017 There are three registration status codes that apply to situations in which a registration
1018 assignment error has been made: DUPLICATE, ANNULLED, and CANCELLED. The
1019 DUPLICATE code refers to the non-surviving record under an exclusivity violation. The
1020 ANNULLED code applied to any other type of assignment error detected *after* an LEI registration
1021 has been publicly issued. The CANCELLED code refers to any type of assignment error detected
1022 *before* publication.

1023 Like RETIRED and MERGED, these error states are terminal: no further updates to registration
1024 records in these states are expected to occur. Records in the DUPLICATE or ANNULLED states
1025 are retained as “Published” records in the GLEIS to support the historical record and to provide
1026 resolution to any external query of the LEI. Records with the state CANCELLED are retained
1027 internally by an LOU to support auditing.

1028 **11.4 Registration Status Codes for Transient Purposes**

1029 The three registration status codes TRANSFERRED, PENDING_TRANSFER, and
1030 PENDING_ARCHIVAL exist to accommodate the transfer of the management of an LEI
1031 registration from one LOU to another.

1032 TRANSFERRED is used in the archival records of a “sending” LOU after having successfully
1033 completed the transfer of that responsibility to another (“receiving”) LOU.

1034 PENDING_TRANSFER and PENDING_ARCHIVAL are used in the transitional period between
1035 the receipt of a porting request by a “sending” pre-LOU and the final confirmation of the
1036 transfer. Particularly in the early stages of the GLEIS, it may happen that for a short time a
1037 given record appears in more than one published file: once in the “receiving” pre-LOU’s
1038 published file with the registration status ISSUED and also in the “sending” pre-LOU’s
1039 published file with registration status PENDING_ARCHIVAL.