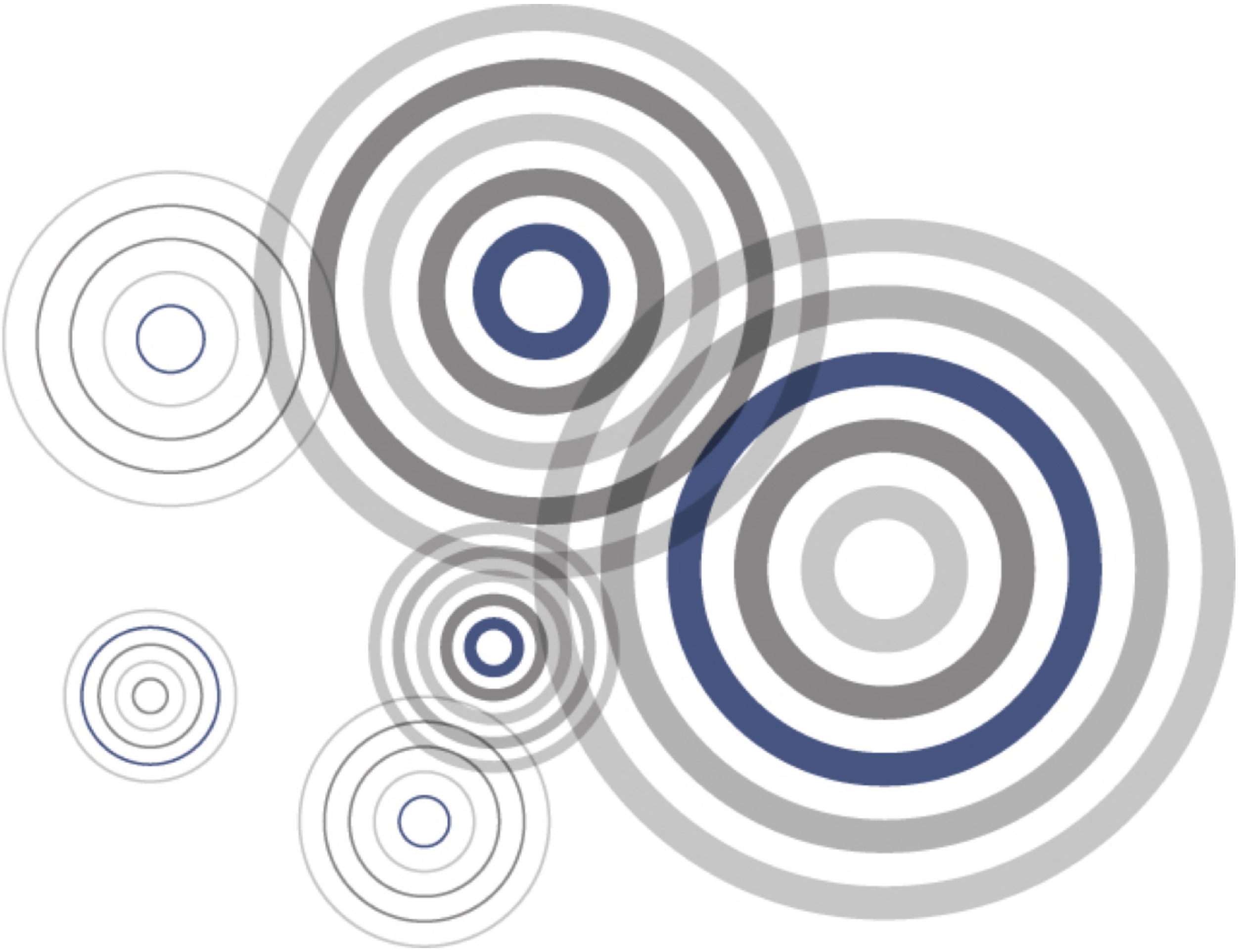
Public

Reporting Exceptions Format V1.1

Final V1.1



Contents

1. Reporting Exceptions Format Version 1.0

Documentation last updated: 2016-11-26

1.1. Introduction

Following the LEI-ROC policy document, "Collecting data on direct and ultimate parents of legal entities in the Global LEI System – Phase 1" (10 March 2016), the Global Legal Identifier System (GLEIS) requires that legal entities with an LEI provide information on their ultimate and direct accounting consolidating parents.

- Relationship reporting according to the Relationship Record Common Data File (RR-CDF) format V1.0 is mandatory. The only opt-out reasons allowed are taken from the LEI-ROC policy document, pages 16-17. A further general exception case, also based on the LEI-ROC policy document (p. 18) is provided to cover situations where the main opt-out reasons may not precisely applicable:
    - No LEI - "the parent does not consent to have an LEI" (LEI-ROC policy, p. 18).
- This format provides a simple record structure linking, **per record:**
    - One LEI from the LOU's current LEI data file;
    - One relationship type (reporting category) that must be reported;
    - One reason for declining to report that relationship type for the legal entity referenced by this LEI, plus an optional reference e.g. to a legal or regulatory provision.

All LOUs use this file format to record and submit Reporting Exceptions to GLEIF.

1.2. Audience for this document

The target audience for this standard includes:

- All Local Operating Units (as well as candidate LOUs) of the GLEIS
- All users or potential users of LEI data
- All financial regulators who consume LEI data

1.3. Status of this document

This section describes the status of this document at the time of its publication. Later versions may supersede this document. The most up to date version will always be available from www.gleif.org
The file format references the LEI ROC's published document entitled "LEI ROC Report on

collecting data on direct and ultimate parents of legal entities in the Global LEI System" (10 March 2016; available from www.leiroc.org).

1.4. Terminology and Typographical Conventions

The following typographical conventions are used throughout the document:

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

1.5. Cardinalities

- The cardinality of each element (the number of times it MUST or may appear in an XML data file conforming to this schema) is expressed as a number range in the format {minimum occurrences, maximum occurrences} in the XML examples shown below the notes of its containing element. This notation is equivalent to the following explanations in words:
  - Mandatory, unique: `{1,1}` - the element MUST appear, exactly once.
  - Mandatory, repeatable: `{1,unbounded}` - the element MUST appear at least once. It may be repeated any number of times.
  - Optional, unique: `{0,1}` - the element NEED NOT appear; it MAY appear once at most.
  - Optional, repeatable: `{0,unbounded}` - the element NEED NOT appear. It MAY be repeated any number of times.

Please note:

- The default cardinality is {1,1} (mandatory, unique). This document highlights when an element differs from this either by its `minOccurs` (minimum occurrences) or `maxOccurs` (maximum occurrences) value, or both.
- XML cardinalities apply in the context of any containing elements. This means that a contained element may have a cardinality of one or more even if its containing element may be omitted, because the contained element is mandatory **given** the presence of the container.
- XML cardinalities enforce a minimum data quality and standards conformance. Other business rules (as explained below) and data quality checks applied by GLEIF may encourage stricter cardinalities in live implementations.

1.6. Business Rules

The accompanying documentation in addition to this Technical Specification specifies business rules where applicable for each element. These are rules that are not enforced by validating against the XML schema, but are still mandatory for all Common Data File (CDF) format files.

1.7. Release Notes

1.7.1. Version 1.1

- Corrections:
  - `Extension` element in `Header` corrected to `minOccurs="0"`.

1.7.2. Version 1.0

The first release.

1.8. Change Management

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

Errata Version An errata version makes corrections to the normative content of the standard (excluding corrections which would change the data schema) and/or makes changes to non-normative content such as explanatory material. An errata version does not change the XML schema definitions, only the documentation parts, and so does not affect the interoperability of systems implementing the standard. An errata version is indicated by incrementing the third version number; e.g., 1.0 to 1.0.1, or 1.0.1 to 1.0.2.

1.8.1. Minor Version

A minor version may include all changes permitted in an errata version, and in addition adds one or more data elements and/or adds one or more codes to a code list ("enum" data type). A minor version changes the XML schema. Minor version changes to schema MUST provide for forward and backward compatibility. This allows existing implementations to continue to interoperate even if they are using different minor versions. A minor version is indicated by incrementing the second version number; e.g., 1.0 to 1.1 or 1.1.3 to 1.2.

1.8.2. Major Version

A major version may make any change at all, including incompatible changes to the XML schema. Major version changes to schema require that the new version uses a different XML namespace. This requires existing implementations to separately understand both the old and new versions during a period of transition. A major version is indicated by incrementing the first version number; e.g., 1.1 to 2.0.

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

1.8.3. Minor Version Changes to the XML Schema

A minor version may introduce new XML elements and/or adds one or more codes to a code list ("enum" data type). Minor version changes to schema SHALL be made as specified below, in order to achieve forward and backward compatibility.

Forward compatibility means that an LEI Data File which is valid according to the older version's schema is also valid according to the newer version's schema.

Backward compatibility means that an LEI Data File which is valid according to the newer version's schema is also valid according to the older version's schema.

New data elements may be added at pre-defined extension points within the schema, each with an optional XML element NextVersion. New data elements are always added within a NextVersion element. When a minor version adds a new data element to a NextVersion element, a new NextVersion element is also added inside the previously added NextVersion element, to accommodate additional data elements in subsequent minor versions. Each successive NextVersion element set is contained directly within the previous minor version's NextVersion set.

As can be seen from the full XML schema presented here, the following rules SHALL be observed to ensure forward and backward compatibility:

- The initial XSD declaration for a NextVersion element SHALL use the element name "NextVersion", XML data type "lei:NextVersion1Type" and cardinality optional, unique {0,1}. The XML data type allows a sequence of any elements, each of cardinality optional, repeatable (unbounded) and with lax content processing, but in the target namespace.

- The minOccurs declaration on the NextVersion element allows it to be omitted in files conforming to the first minor version. The schema wildcard xsd:any allows for forward compatibility: a file conforming to a new minor version still validates in the old version because the wildcard matches any new elements introduced in the new minor version.
- New elements SHALL be introduced in a subsequent minor version by modifying the declaration for the above type declaration as follows:
  - A sequence of the new elements introduced in the previous version
  - A subsequent NextVersionN element where N is an index number starting at 1 and incremented by 1 with each minor version
- Each new element SHALL be declared minOccurs=”0”, to ensure backward compatibility: a file conforming to the old version still validates in the new version because the new schema does not require the presence of elements not defined in the old version. If a new element is mandatory for conformance to the new version, this MUST be enforced outside schema validation.
- The new definition of the NextVersion element SHALL include a declaration of an inner NextVersion element, as illustrated above, to provide for additional elements in subsequent minor versions. The nesting of NextVersion elements is required to satisfy the “unique particle attribution” constraint of XSD 1.0.
- Each code list (Enum types) is implemented in the XML schema simply as the XSD string data type. This provides for forward compatibility because the schema for an older minor version will validate any string, including codes defined in newer minor versions. The schema for each minor version includes the list of valid codes for that minor version as a documentation annotation to the type declaration for each Enum type.

1.8.4. Major Version Changes to the XML Schema

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

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

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

1.9. XML Syntax

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

1.9.1. XML Design Rules

- The XSD schema conforms to W3C's XML Schema specification, version 1.0.
- The XML namespace is "http://www.gleif.org/data/schema/repex/2016".
- All interior XML elements are namespace-qualified (element form = qualified).
- All XML attributes are in the null namespace (attribute form = unqualified), with the exception of `xml:lang`.
- Element names are upper camel case.
- Attribute name are lower camel case.
- XSD type names are upper camel case.
- Enumeration code list values are all caps with underscores.
- Elements are used in preference to attributes except for language and type qualifiers.
- For a data element specified as having unbounded cardinality, the XML includes a single container element whose subelements are one or more instances of the data element whose cardinality is unbounded. The name of the container element is formed as the plural of the name of the contained elements.

1.9.2. XML Schema

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

```
<?xml version="1.0" encoding="utf-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"

    xmlns:repex="http://www.gleif.org/data/schema/repex/2016"
elementFormDefault="qualified"

    targetNamespace="http://www.gleif.org/data/schema/repex/2016">

    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
```

```
        schemaLocation="http://www.w3.org/2001/xml.xsd"/>
    <xs:element name="ReportingExceptionData"
type="repex:ReportingExceptionDataType"/>
    <xs:complexType name="ReportingExceptionDataType">
        <xs:sequence>
            <xs:element name="Header" type="repex:ExceptionHeaderType">
</xs:element>
            <xs:element name="ReportingExceptions"
type="repex:ReportingExceptionsType"
            > </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ExceptionHeaderType">
        <xs:sequence>
            <xs:element name="ContentDate" type="repex:LEIDateTimeProfile">
</xs:element>
            <xs:element name="Originator" type="repex:LEIType"
minOccurs="0"> </xs:element>
            <xs:element name="FileContent" type="repex:FileContentEnum">
</xs:element>
            <xs:element name="DeltaStart" type="repex:LEIDateTimeProfile"
minOccurs="0"> </xs:element>
            <xs:element name="RecordCount" type="xs:nonNegativeInteger">
</xs:element>
            <xs:element name="NextVersion"
type="repex:HeaderNextVersionType" minOccurs="0"/>
            <xs:element name="Extension" type="repex:ExtensionType"
minOccurs="0"> </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="LEIType">
        <xs:restriction base="xs:string">
            <xs:minLength value="20"/>
            <xs:maxLength value="20"/>
            <xs:pattern value="([0-9A-Z]{18}[0-9]{2})"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="FileContentEnum">
        <xs:restriction base="xs:string">
```

```
            <xs:enumeration value="LOU_FULL_PUBLISHED"> </xs:enumeration>
            <xs:enumeration value="LOU_DELTA_PUBLISHED"> </xs:enumeration>
            <xs:enumeration value="GLEIF_FULL_PUBLISHED"> </xs:enumeration>
            <xs:enumeration value="GLEIF_DELTA_PUBLISHED">
</xs:enumeration>
            <xs:enumeration value="QUERY_RESPONSE"> </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="ExtensionType">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax" namespace="##other"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ExceptionType">
        <xs:sequence>
            <xs:element name="LEI" type="repex:LEIType"> </xs:element>
            <xs:element name="ExceptionCategory"
type="repex:ExceptionCategoryEnum"> </xs:element>
            <xs:element name="ExceptionReason"
type="repex:ExceptionReasonEnum"
                maxOccurs="unbounded"> </xs:element>
            <xs:element name="ExceptionReference"
type="repex:Tokenized500Type" minOccurs="0"
                maxOccurs="unbounded"> </xs:element>
            <xs:element name="NextVersion"
type="repex:ReportingExceptionNextVersionType"
                minOccurs="0"/>
            <xs:element name="Extension" type="repex:ExtensionType"
minOccurs="0"> </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ReportingExceptionsType">
        <xs:sequence>
            <xs:element name="Exception" type="repex:ExceptionType"
minOccurs="0"
                maxOccurs="unbounded"> </xs:element>
            <xs:element name="NextVersion"
type="repex:ReportingExceptionsContainerNextVersionType"
```

```
                minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="ExceptionCategoryEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="DIRECT_ACCOUNTING_CONSOLIDATION_PARENT">
</xs:enumeration>
            <xs:enumeration
value="ULTIMATE_ACCOUNTING_CONSOLIDATION_PARENT"> </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ExceptionReasonEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="NO_LEI"> </xs:enumeration>
            <xs:enumeration value="NATURAL_PERSONS"> </xs:enumeration>
            <xs:enumeration value="NON_CONSOLIDATING"> </xs:enumeration>
            <xs:enumeration value="NO_KNOWN_PERSON"> </xs:enumeration>
            <xs:enumeration value="LEGAL_OBSTACLES"> </xs:enumeration>
            <xs:enumeration value="CONSENT_NOT_OBTAINED"> </xs:enumeration>
            <xs:enumeration value="BINDING_LEGAL_COMMITMENTS">
</xs:enumeration>
            <xs:enumeration value="DETRIMENT_NOT_EXCLUDED">
</xs:enumeration>
            <xs:enumeration value="DISCLOSURE_DETRIMENTAL">
</xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Tokenized500Type">
        <xs:restriction base="xs:string">
            <xs:maxLength value="500"/>
            <xs:minLength value="1"/>
            <xs:pattern value="\S+( \S+)*"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="LEIDateTimeProfile">
        <xs:restriction base="xs:dateTime">
            <xs:pattern
```

```
                value="([^\.]*|([^\.]*(\.((\d){1,3})){0,1}))(Z|\+([01][0-
9]|2[0-3]):([0-5][0-9])|-([01][0-9]|2[0-3]):([0-5][0-9]))"
            />
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="ReportingExceptionsContainerNextVersionType">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"
                namespace="##targetNamespace"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ReportingExceptionNextVersionType">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"
                namespace="##targetNamespace"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="HeaderNextVersionType">
        <xs:sequence>
            <xs:any minOccurs="0" maxOccurs="unbounded"
processContents="lax"
                namespace="##targetNamespace"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

1.9.3. Extension

The optional Extension section of a Reporting Exception Item may be used to include additional data not defined in this standard. This may include data specific to an LOU, data specific to a publisher of LEI data, and so on.

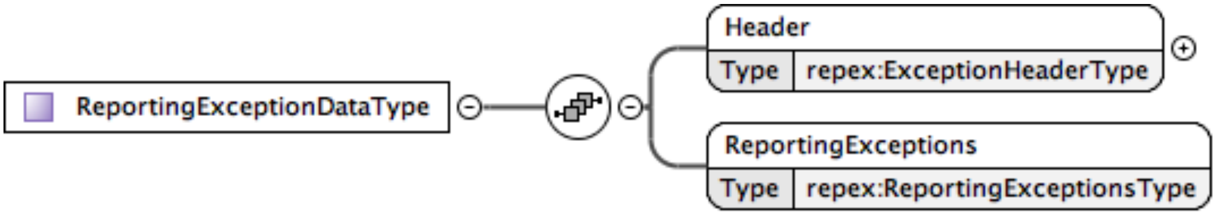
For example, an LOU may use Extension to publish additional data elements it collects as part of registration.

The following rules MUST be observed:

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

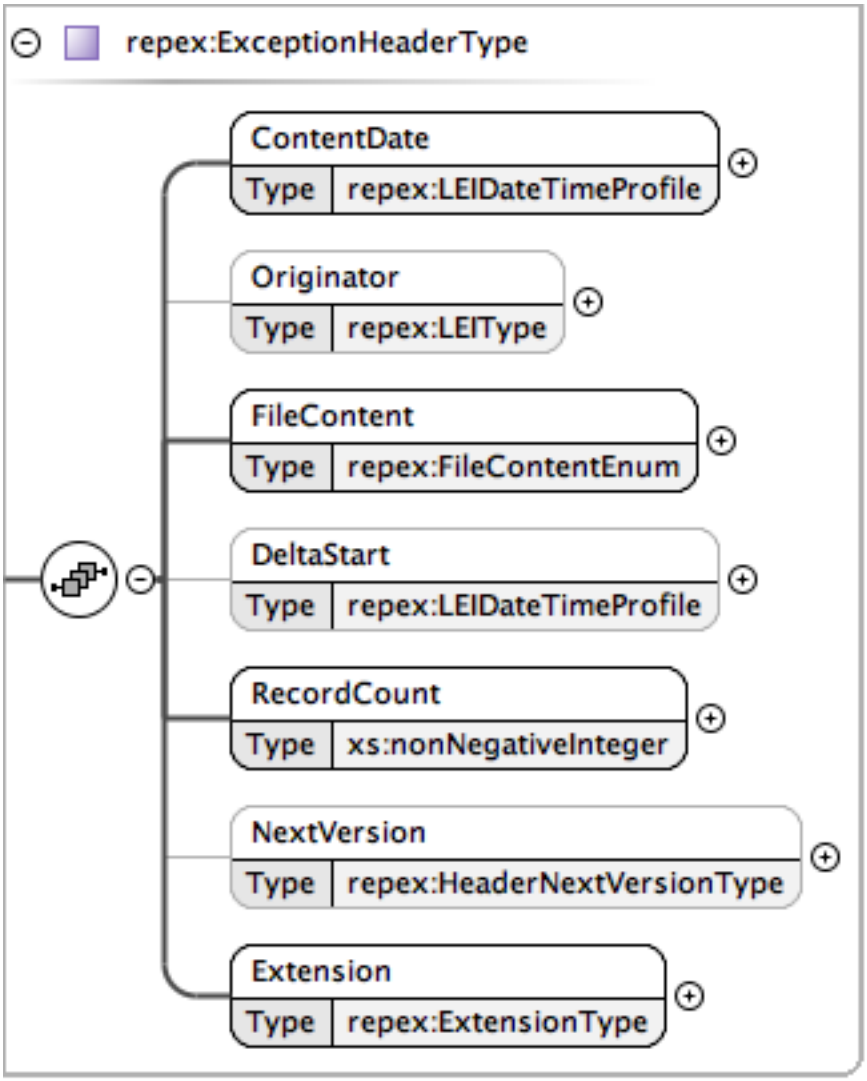
2. Abstract Data Content

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

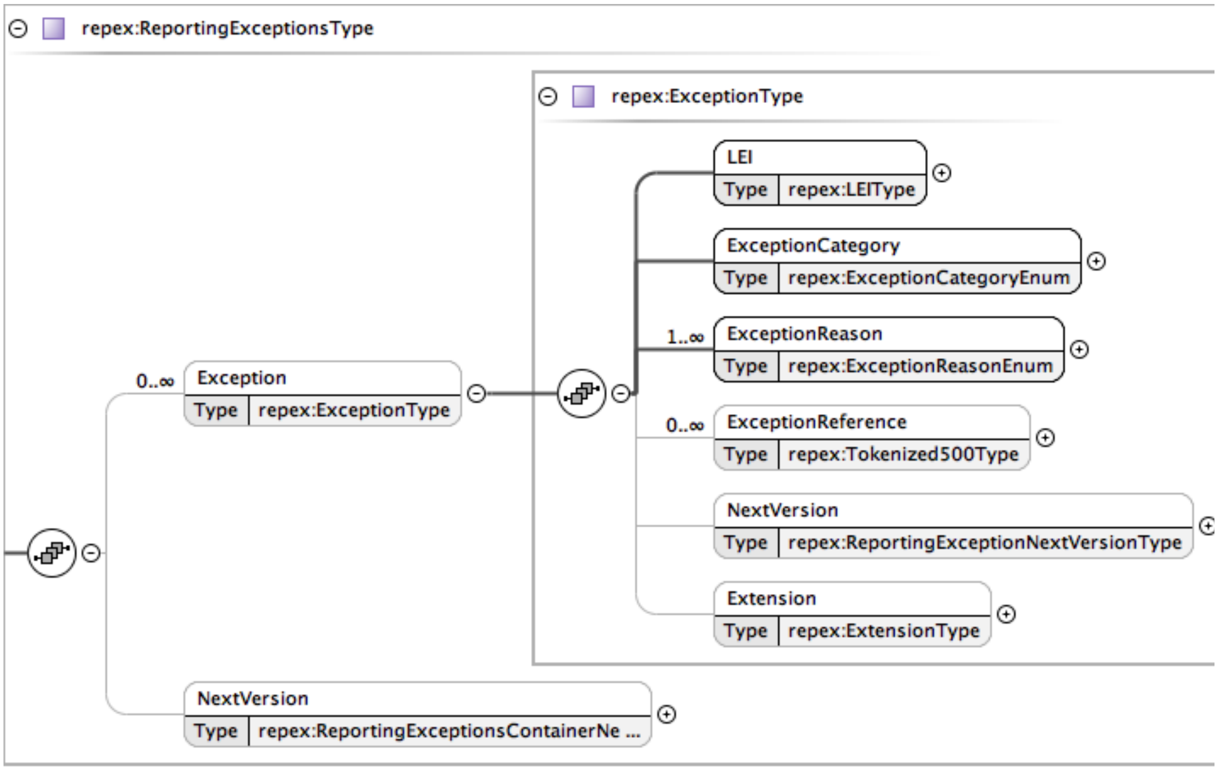

ReportingExceptionDataType
Header
Type
repex:ExceptionHeaderType
ReportingExceptions
Type
repex:ReportingExceptionsType

A data file conforming to this standard SHALL consist of:

- A Header:


repex:ExceptionHeaderType
ContentDate
Type repex:LEIDateTimeProfile
Originator
Type repex:LEIType
FileContent
Type repex:FileContentEnum
DeltaStart
Type repex:LEIDateTimeProfile
RecordCount
Type xs:nonNegativeInteger
NextVersion
Type repex:HeaderNextVersionType
Extension
Type repex:ExtensionType

- Zero or more Reporting Exception Items:


repex:ReportingExceptionsType
repex:ExceptionType
LEI
Type repex:LEIType
ExceptionCategory
Type repex:ExceptionCategoryEnum
1..∞
ExceptionReason
Type repex:ExceptionReasonEnum
0..∞
Exception
Type repex:ExceptionType
0..∞
ExceptionReference
Type repex:Tokenized500Type
NextVersion
Type repex:ReportingExceptionNextVersionType
Extension
Type repex:ExtensionType
NextVersion
Type repex:ReportingExceptionsContainerNe ...

2.1. Data Element Reference

All elements, attributes and datatypes in the format are listed below, one per page.

2.1.1. Element repex:ReportingExceptionData

repex:ReportingExceptionDataType

content **complex**

```
<repex:ReportingExceptionData
xmlns:repex="http://www.gleif.org/data/schema/repex/2016">
  <repex:Header>{1,1}</repex:Header>
  <repex:ReportingExceptions>{1,1}</repex:ReportingExceptions>
</repex:ReportingExceptionData>
```

2.1.2. Element repex:ReportingExceptionDataType / repex:Header

Contains the file upload information for this Reporting Exceptions file

repex:ExceptionHeaderType

content **complex**

```
<repex:Header xmlns:repex="http://www.gleif.org/data/schema/repex/2016">
  <repex:ContentDate>{1,1}</repex:ContentDate>
  <repex:Originator>{0,1}</repex:Originator>
  <repex:FileContent>{1,1}</repex:FileContent>
  <repex:DeltaStart>{0,1}</repex:DeltaStart>
  <repex:RecordCount>{1,1}</repex:RecordCount>
  <repex:NextVersion>{0,1}</repex:NextVersion>
  <repex:Extension>{0,1}</repex:Extension>
</repex:Header>
```

2.1.3. Element repex:ExceptionHeaderType / repex:ContentDate

The date and time as of which the data contained in the file is valid.

repex:LEIDateTimeProfile

content	**simple**
pattern	**([^\\.]*|([^\\.]*(\\.((\\d){1,3})){0,1}))(Z|\\+([01][0-9]|2[0-3]):([0-5][0-9])|-([01][0-9]|2[0-3]):([0-5][0-9]))**

2.1.4. Element repex:ExceptionHeaderType / repex:Originator

The LEI of the entity that created the content of this file.

repex:LEIType

content	**simple**
minOccurs	**0**

minLength	**20**
maxLength	**20**
pattern	**([0-9A-Z]{18}[0-9]{2})**

2.1.5. Element repex:ExceptionHeaderType / repex:FileContent

A code describing the content of this RelationshipRecords file.

repex:FileContentEnum

content **simple**

enumeration **LOU_FULL_PUBLISHED**

The file contains all Reporting Exceptions published by an LOU (all Reporting Exceptions for which the LOU is the `ManagingLOU`) as of the date/time the file is created.

enumeration **LOU_DELTA_PUBLISHED**

The file contains those Reporting Exceptions published by an LOU (all Reporting Exceptions for which the LOU is the `ManagingLOU`) which are new or changed since the `DeltaStart` specified in the header, as of the date/time the file is created.

enumeration **GLEIF_FULL_PUBLISHED**

The file contains all Reporting Exceptions published by GLEIF (including all Reporting Exceptions from all LOUs) as of the date/time the file is created.

enumeration **GLEIF_DELTA_PUBLISHED**

The file contains those Reporting Exceptions published by GLEIF (including all Reporting Exceptions from all LOUs) which are new or changed since the `DeltaStart` date specified in the `Header`, as of the date/time the file is created.

enumeration **QUERY_RESPONSE**

The file contains records matching criteria specified in a query.

2.1.6. Element repex:ExceptionHeaderType / repex:DeltaStart

The date and time of the baseline relative to which this file contains new or changed Reporting Exceptions.

repex:LEIDateTimeProfile

<table>
<tr><td>content</td><td>simple</td></tr>
<tr><td>minOccurs</td><td>0</td></tr>
<tr><td>pattern</td><td>([^\.]*|([^\.]*(\.((\d){1,3})){0,1}))(Z|\+([01][0-9]|2[0-3]):([0-5][0-9])|-([01][0-9]|2[0-3]):([0-5][0-9]))</td></tr>
</table>

2.1.7. Element repex:ExceptionHeaderType / repex:RecordCount

The number of Reporting Exceptions (RepEx) in the file. Must be a positive whole (integer) number, or zero (0).

xs:nonNegativeInteger

content **simple**

2.1.8. Element repex:ExceptionHeaderType / repex:NextVersion

repex:HeaderNextVersionType

content	**complex**
minOccurs	**0**

2.1.9. Element repex:ExceptionHeaderType / repex:Extension

This Extension element contains any additional elements required to extend the Header container element.

repex:ExtensionType

content	**complex**
minOccurs	**0**

2.1.10. Element repex:ReportingExceptionDataType / repex:ReportingExceptions

A list of reporting exceptions, with reason(s) for each exception, provided by legal entities, for declining to provide specified classes of relationship information

repex:ReportingExceptionsType

content **complex**

```
<repex:ReportingExceptions
xmlns:repex="http://www.gleif.org/data/schema/repex/2016">
  <repex:Exception>{0,unbounded}</repex:Exception>
  <repex:NextVersion>{0,1}</repex:NextVersion>
</repex:ReportingExceptions>
```

2.1.11. Element repex:ReportingExceptionsType / repex:Exception

A single exception to a specified reporting requirement, giving reasons and references where applicable.

repex:ExceptionType

content	**complex**
minOccurs	**0**
maxOccurs	**unbounded**

```
<repex:Exception
xmlns:repex="http://www.gleif.org/data/schema/repex/2016">
  <repex:LEI>{1,1}</repex:LEI>
  <repex:ExceptionCategory>{1,1}</repex:ExceptionCategory>
  <repex:ExceptionReason>{1,unbounded}</repex:ExceptionReason>
  <repex:ExceptionReference>{0,unbounded}</repex:ExceptionReference>
  <repex:NextVersion>{0,1}</repex:NextVersion>
  <repex:Extension>{0,1}</repex:Extension>
</repex:Exception>
```

2.1.12. Element repex:ExceptionType / repex:LEI

The ISO 17442 compatible identifier for the legal entity which raised this exception.

repex:LEIType

content **simple**

minLength	**20**
maxLength	**20**
pattern	**([0-9A-Z]{18}[0-9]{2})**

2.1.13. Element repex:ExceptionType / repex:ExceptionCategory

A category of mandatory information reporting, beyond the minimum required for legal entity identification, which the legal entity declines, giving reasons and references where applicable.

repex:ExceptionCategoryEnum

content **simple**

enumeration **DIRECT_ACCOUNTING_CONSOLIDATION_PARENT**	The legal entity has declined to report a direct accounting consolidation parent, based on applicable accounting standards.
enumeration **ULTIMATE_ACCOUNTING_CONSOLIDATION_PARENT**	The legal entity has declined to report an ultimate accounting consolidation parent, based on applicable accounting standards.

2.1.14. Element repex:ExceptionType / repex:ExceptionReason

A single reason provided by the legal entity for declining to provide the mandatory report of a specified type of information (beyond the minimum reference data needed purely for identification of the legal entity).

repex:ExceptionReasonEnum

content **simple**

maxOccurs **unbounded**

enumeration **NO_LEI**

The parent does not consent to have an LEI.

enumeration **NATURAL_PERSONS**

There is no parent according to the definition used, because the entity is controlled by natural person(s) without any intermediate legal entity meeting the definition of accounting consolidating parent.

enumeration **NON_CONSOLIDATING**

There is no parent according to the definition used, because the entity is controlled by legal entities not subject to preparing consolidated financial statements.

enumeration **NO_KNOWN_PERSON**

There is no parent according to the definition used, because there is no known person controlling the entity (e.g.., diversified shareholding).

enumeration **LEGAL_OBSTACLES**

Obstacles in the laws or regulations of a jurisdiction prevent providing or publishing this information. This does not include cases where, under the applicable legal framework disclosing the parent relationship would require the consent of

	one of the entities in the relationship, or both, and such consent could not be obtained (in these cases "CONSENT_NOT_OBTAINED" is the applicable code). The LOU is not expected to verify or analyze whether the legal framework constitutes a legal obstacle.
enumeration **CONSENT_NOT_OBTAINED**	Obstacles in the laws or regulations of a jurisdiction prevent providing or publishing this information: “the consent of the parent was necessary under the applicable legal framework and the parent did not consent or could not be contacted”. Note that it is a responsibility of a child entity to seek parent consent when necessary for disclosing the parent relationship, for instance by inviting in writing the parent entity to provide consent. The LOU is not expected to verify or analyze whether the legal framework constitutes a legal obstacle.
enumeration **BINDING_LEGAL_COMMITMENTS**	Binding legal commitments (other than the laws or regulations of a jurisdiction), such as articles governing the legal entity or a contract, prevent providing or publishing this information. The LOU is not expected to verify or analyze whether the legal framework constitutes a legal obstacle.
enumeration **DETRIMENT_NOT_EXCLUDED**	The child entity has sought to consult the parent entity about the reporting of the parent information to the GLEIS but could not confirm the absence of detriment in a way that can appropriately prevent liability risks for the child entity (or those acting on its behalf) under the applicable legal framework. The disclosure of this information would be detrimental to the legal entity or the relevant parent. This will include reasons generally accepted by

	public authorities in similar circumstances, based on a declaration by the entity. This reason may be used only when all following cumulative circumstances apply: [i) the parent could not be informed via the GLEIS and have the possibility to correct the relationship information before publication (including raising a cause for opt out, either because the parent does not have an LEI, or it has an LEI but the GLEIS has not yet implemented such system;] ii) the relationship is not already in the public domain (information being in the public domain assumes here that the way the information came into the public domain did not infringe the applicable legal framework); iii) the child entity has reasons to believe that the parent may consider disclosure to be detrimental; (iv) the child entity has sought to consult the parent entity of the reporting of the parent information to the GLEIS but could not confirm the absence of detriment in a way that can appropriately prevent liability risks for the child entity (or those acting on its behalf) under the applicable legal framework.
enumeration **DISCLOSURE_DETRIMENTAL**	The disclosure of this information would be detrimental to the legal entity or the relevant parent. This will include reasons generally accepted by public authorities in similar circumstances, based on a declaration by the entity.

2.1.15. Element repex:ExceptionType / repex:ExceptionReference

References of the law, regulation or other element of the legal framework to support reason(s) provided by the legal entity for declining to provide information on its parents.

repex:Tokenized500Type

content	**simple**
minOccurs	**0**
maxOccurs	**unbounded**

minLength	**1**
maxLength	**500**
pattern	**\S+(\S+)***

2.1.16. Element repex:ExceptionType / repex:NextVersion

repex:ReportingExceptionNextVersionType

content	**complex**
minOccurs	**0**

2.1.17. Element repex:ExceptionType / repex:Extension

This `repex:Extension` element may contain any additional elements required to extend the Reporting Exception.

repex:ExtensionType

content	**complex**
minOccurs	**0**

2.1.18. Element repex:ReportingExceptionsType / repex:NextVersion

repex:ReportingExceptionsContainerNextVersionType

content	**complex**
minOccurs	**0**

2.1.19. Simple Type repex:LEIDateTimeProfile

<table>
<tr><td>restriction of **xs:dateTime**</td><td></td></tr>
<tr><td>pattern</td><td>**([^\.]*|([^\.]*(\.((\d){1,3})){0,1}))(Z|\+([01][0-9]|2[0-3]):([0-5][0-9])|-([01][0-9]|2[0-3]):([0-5][0-9]))**</td></tr>
</table>

2.1.20. Simple Type repex:LEIType

restriction of **xs:string**

minLength	**20**
maxLength	**20**
pattern	**([0-9A-Z]{18}[0-9]{2})**

2.1.21. Simple Type repex:FileContentEnum

restriction of **xs:string**

enumeration **LOU_FULL_PUBLISHED**	The file contains all Reporting Exceptions published by an LOU (all Reporting Exceptions for which the LOU is the `ManagingLOU`) as of the date/time the file is created.
enumeration **LOU_DELTA_PUBLISHED**	The file contains those Reporting Exceptions published by an LOU (all Reporting Exceptions for which the LOU is the `ManagingLOU`) which are new or changed since the `DeltaStart` specified in the header, as of the date/time the file is created.
enumeration **GLEIF_FULL_PUBLISHED**	The file contains all Reporting Exceptions published by GLEIF (including all Reporting Exceptions from all LOUs) as of the date/time the file is created.
enumeration **GLEIF_DELTA_PUBLISHED**	The file contains those Reporting Exceptions published by GLEIF (including all Reporting Exceptions from all LOUs) which are new or changed since the `DeltaStart` date specified in the `Header`, as of the date/time the file is created.
enumeration **QUERY_RESPONSE**	The file contains records matching criteria specified in a query.

2.1.22. Simple Type repex:ExceptionCategoryEnum

restriction of **xs:string**

enumeration **DIRECT_ACCOUNTING_CONSOLIDATION_PARENT**	The legal entity has declined to report a direct accounting consolidation parent, based on applicable accounting standards.
enumeration **ULTIMATE_ACCOUNTING_CONSOLIDATION_PARENT**	The legal entity has declined to report an ultimate accounting consolidation parent, based on applicable accounting standards.

2.1.23. Simple Type repex:ExceptionReasonEnum

Type	restriction of **xs:string**	
Facets	enumeration **NO_LEI**	The parent does not consent to have an LEI.
	enumeration **NATURAL_PERSONS**	There is no parent according to the definition used, because the entity is controlled by natural person(s) without any intermediate legal entity meeting the definition of accounting consolidating parent.
	enumeration **NON_CONSOLIDATING**	There is no parent according to the definition used, because the entity is controlled by legal entities not subject to preparing consolidated financial statements.
	enumeration **NO_KNOWN_PERSON**	There is no parent according to the definition used, because there is no known person controlling the entity (e.g.., diversified shareholding).
	enumeration **LEGAL_OBSTACLES**	Obstacles in the laws or regulations of a jurisdiction prevent providing or publishing this information. This does not include cases where, under the applicable legal framework disclosing the parent relationship would require the consent of one of the entities in the relationship, or both, and such consent could not be obtained (in these cases "CONSENT_NOT_OBTAINED" is the applicable code). The LOU is not expected to verify or analyze

	whether the legal framework constitutes a legal obstacle.
enumeration **CONSENT_NOT_OBTAINED**	Obstacles in the laws or regulations of a jurisdiction prevent providing or publishing this information: “the consent of the parent was necessary under the applicable legal framework and the parent did not consent or could not be contacted”. Note that it is a responsibility of a child entity to seek parent consent when necessary for disclosing the parent relationship, for instance by inviting in writing the parent entity to provide consent. The LOU is not expected to verify or analyze whether the legal framework constitutes a legal obstacle.
enumeration **BINDING_LEGAL_COMMITMENTS**	Binding legal commitments (other than the laws or regulations of a jurisdiction), such as articles governing the legal entity or a contract, prevent providing or publishing this information. The LOU is not expected to verify or analyze whether the legal framework constitutes a legal obstacle.
enumeration **DETRIMENT_NOT_EXCLUDED**	The child entity has sought to consult the parent entity about the reporting of the parent information to the GLEIS but could not confirm the absence of detriment in a way that can appropriately prevent liability risks for the child entity (or those acting on its behalf) under the applicable legal framework. The disclosure of this information would

	be detrimental to the legal entity or the relevant parent. This will include reasons generally accepted by public authorities in similar circumstances, based on a declaration by the entity. This reason may be used only when all following cumulative circumstances apply: [i) the parent could not be informed via the GLEIS and have the possibility to correct the relationship information before publication (including raising a cause for opt out, either because the parent does not have an LEI, or it has an LEI but the GLEIS has not yet implemented such system;] ii) the relationship is not already in the public domain (information being in the public domain assumes here that the way the information came into the public domain did not infringe the applicable legal framework); iii) the child entity has reasons to believe that the parent may consider disclosure to be detrimental; (iv) the child entity has sought to consult the parent entity of the reporting of the parent information to the GLEIS but could not confirm the absence of detriment in a way that can appropriately prevent liability risks for the child entity (or those acting on its behalf) under the applicable legal framework.
enumeration **DISCLOSURE_DETRIMENTAL**	The disclosure of this information would be detrimental to the legal entity or the relevant parent. This will include reasons generally

	accepted by public authorities in similar circumstances, based on a declaration by the entity.

2.1.24. Simple Type repex:Tokenized500Type

An element of this type has minimum length of one character and may not contain any of: the carriage return (#xD), line feed (#xA) nor tab (#x9) characters, shall not begin or end with a space (#x20) character, or a sequence of two or more adjacent space characters.

restriction of **xs:string**

minLength	**1**
maxLength	**500**
pattern	**\S+(\S+)***