EBA/XBRL/2015/01

10 February 2015

EBA XBRL Filing Rules

Contents

Change History

Version	Date	Changes
1		Baseline
2	March 2014	Included missing bibliographic references Reordered auxiliary sections Slight expansion of rules around filing indicators, and inclusion of illustrative examples Further elaboration of the scope of applicability of these rules, highlighting discretion of the competent regulatory authorities as to format and mechanism of reporting (i.e. EBA XBRL not compulsory at first level reporting). Added requirement for pre-registered LEI code to be used as entity identifier in 2nd level remittance, and recommendation of scheme URI to use for LEI (and other) entity codes Emphasize that @xml:lang is not generally required by EBA
3	February 2015	Rule 1.6 altered to indicate requirement to include negative filing indicators to indicate non-reporting (“nil” report) for expected templates, in accordance with new business instructions. Minor tweaks to other text referring to filing indicators to clarify where “positive” indicators are being discussed. Annotated instructions regarding monetary values to highlight possibility of explicitly being requested to report monetary values as decimals (without currency units), and resultant effects. *Note that it is considered somewhat likely that rule 3.1 (requiring only a single explicit currency to be reported per instance) may need to be relaxed in future (i.e. if required by future EBA reporting requirements).* Improved layout and phrasing in table at rule 2.19. Wording improvement and removal of comment regarding @decimals and @precision being used on the same fact (which is anyway contrary to XBRL 2.1 spec and so invalid XBRL). Remove reference to MFI ID, or specific national IDs from 3.6

Abbreviations

UML	Unified Modeling Language
W3C	World Wide Web Consortium
XBRL	eXtensible Business Reporting Language
XML	eXtensible Markup Language

Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

XBRL 2.1
XBRL Dimensions 1.0
XBRL Registry specification 1.0
XBRL Formula specification 1.0

Bibliography

[CWA] CEN Workshop Agreement “European Filing Rules” working draft[1]

[GFM11] Global Filing Manual (Interoperable Taxonomy Architecture Project)

[EFM13] EDGAR Filer Manual. U.S. Securities and Exchange Commission

[FRIS04] Financial Reporting Instance Standards 1.0

[SBR13] SBR FRIS rules 2013

[1] draft as of 19/11/2013, see www.wikixbrl.info/index.php?title=European_Filing_Rules

Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE XBRL specific terms like context, unit, period, entity, s-equal, v-equal see XBRL 2.1

applicable taxonomy

an XBRL taxonomy recognised to use as a base for filings in a given filing system

data point

a Data Point is an information component that is defined by a supervisory authority to be sent in an instance document

Note: In XBRL a data point is represented by a fact, its value and related dimensional combinations.

dimension

a Dimension is an xs:element in the substitutionGroup of xbrldt:dimensionItem; it relates to the ability to express multidimensional information

entrypoint

a schema or linkbase in the applicable taxonomy that represents the filing requirements and gets mentioned in the instance by the filer.

fact

a fact is an occurrence in an instance document of an element with a mandatory contextRef attribute and optional attributes like unitRef, xml:lang or xsi:nil

filer

an entity responsible for submission of a filing.

filing

a filing is the fundamental unit of information that is transmitted to a filing system for receipt, validation and acceptance

Note: a filing is conveyed in an XBRL instance document or series of XBRL instance documents.

filing system

a system in which XBRL instance documents are filed, received, analysed and redistributed

reporter

a reporting entity – described by instance(s)

reporting unit

set of facts in a filing which are conceptually either reported or not reported together as a unit.

template

a (usually tabular) visible representation of a set of facts, typically identified with/as a single reporting unit.

Introduction

The eXtensible Business Reporting Language (XBRL) specification provides a high degree of flexibility in the creation of XBRL instance documents. Part of this flexibility stems from the nature of the syntax: XML, and part stems from the XBRL specification itself.

Scope of Application

The European supervisory reporting process is conceptually a two stage process, first institutions report supervisory data to their relevant national or supranational regulatory authorities ("first level reporting"), and subsequently those authorities remit that data to the European Banking Authority ("second level reporting").

These filing rules represent a collection of additional rules and guidance specifically **applicable to the second level remittance** of XBRL instances for regulatory filings of COREP and FINREP **by relevant national and supranational authorities to the European Banking Authority**. These rules constrain the full flexibility of XBRL, to enable effective interaction between transmitter and recipient/consumer of regulatory reports.

The listed filing rules are influenced by the EBA Taxonomy Architecture in cases where the instance creation is affected.

N.B these rules are not necessarily those that are applicable at the level of reporting by individual institutions or groups of institutions. Guidance should be sought from the reporter's competent authority as to their reporting format and requirements for that reporting.

Basis in harmonized "European Filing Rules" guidance

In order to promote and enhance interoperability, these rules are largely drawn from the CEN Workshop Agreement "European Filing Rules" working draft document (as of 19/11/2013, see www.wikixbrl.info/index.php?title=European_Filing_Rules), which *"represent a collection of recommendations to be seen as guidance to be implemented in the European supervisory reporting process"*. This draft should be read in conjunction/comparison with that document.

For ease of comparison, rules are numbered as per the CEN document (hence some numbers are omitted where the corresponding CEN rule is not applicable/not included, and some additional rules are inserted.

Target Audience

This document is intended for a technical audience and assumes that the reader has a working knowledge of the XBRL 2.1 and the XBRL Dimensions 1.0 Specifications and has a basic understanding of XML, Namespaces, and XML Schema.

To readers with XML knowledge, many of the guidelines in this document will be familiar however, others originate from features that are XBRL-specific and therefore the reasoning behind them may be less obvious.

To ease the understanding by software developers implementing these guidelines in their reporting system, an UML model is included to show the relationships between the different XBRL objects mentioned in this document.

Some of the filing rules are accompanied by constraints defined in the Object Constraint Language (OCL). OCL is part of the UML and allows description of constraints based on the UML objects of the class model. OCL is not a programming language; it just supports the definition of technical specifications. OCL eases the understanding of the rules by using a formal language to provide an unambiguous and consistent description.

XML attribute names are preceded by the "@" character in this document, as in XPath syntax.

Relationship to Other Work

The guidelines in this document pertain to XBRL filings. Parts of this document reiterate for expository clarity certain syntactic and semantic restrictions imposed by XBRL, but this document does not modify XBRL. In the event of any conflicts between this document and XBRL, XBRL prevails. This document does place additional restrictions beyond those prescribed by XBRL.

The rules are based closely on the recommendations of the CEN Workshop Agreement on European filing rules developed by the CEN WS/XBRL project (http://cen.eurofiling.info/).

For harmonization and explanatory purposes, where similar filing rules are used in other jurisdictions, references are indicated.

Use of Language

In the rules that follow, the use of “MUST” implies an obligation, and the preparation of instance files not following these rules will generally result in rejection of the instance file.

The use of “SHOULD” implies an indication of preference or best practice, but also a degree of tolerance, following the principle of “comply or explain”). The rule must be respected unless there are good reasons not to do so. Failure to follow the rule will in general not result in rejection of an instance file.

The use of “MAY” implies permission, and describes actions that can be taken or constructs that can be used. Utilising these options will not result in rejection of an instance file.

1. Filing syntax rules

1.1 — Filing naming

Common practice is to use the extension .xbrl for instance documents. Detailed file naming requirements should be confirmed with the intended recipient of an instance file (i.e. credit institutions should confirm with their relevant competent authority for reporting, CAs should confirm with the EBA for remittance).

1.4 — Character encoding of XBRL instance documents

The XML and XBRL specifications place no restrictions on the character encodings that may be used in instance documents. In order to avoid using a character encoding that is not supported by a receiving processor, all instances should use the UTF-8 character encoding.

XBRL instance documents MUST use "UTF-8" encoding. [GFM11, p. 11]

```
context xmlDocument inv: self.encoding = 'UTF-8'
```

1.5 — Taxonomy entry point selection

A taxonomy is loaded through a reference to one or more URLs, with other files in the taxonomy being included through the process of DTS Discovery. Although technically a user can reference any file in the taxonomy, a taxonomy publisher will typically nominate specific URLs which are intended to be referenced by users of the taxonomy. These URLs are called entry points, and allow users to import the correct modules from the taxonomy, with different modules including different templates and different associated validation rules.

The EBA taxonomy defines multiple specific entry points (“modules”), suitable for different reports. The taxonomy also contains multiple tables, these are not to be treated as entry points. Through the 'filing indicators' it is communicated which tables are reported in an instance.

(a) Reporting entities MUST reference only one entry point schema (“module”, link:schemaRef element), as specified in the applicable taxonomy, per XBRL instance. [SBR13, p. 6]

(b) The schemaRef element MUST refer to a URI appropriate to the reference date of an instance, drawn from the list of entry points published by the EBA[2].

[2] or competent authority for first level reporting.

1.6 — Filing indicators

Each reported fact in a filing is assigned to one or more reporting units (also known as "templates") of the specific domain of reporting.

A filing indicator element (filingIndicator), grouped (potentially with other such elements) within a containing element (fIndicators), containing a code associated with a particular reporting unit, is used to indicate the intention of a reporter to report that reporting unit, or to indicate the intention *not* to report that reporting unit (see example under the heading "Filing indicator examples" for illustration). Filing indicators also trigger the appropriate taxonomy formulae checks. Missing filing indicators can lead to inconsistencies because facts for unindicated reporting units might not be validated.

(a) Reported XBRL instances MUST include appropriate positive (i.e. either with @filed="true" or without an @filed attribute) filing indicator elements to express which reporting units ("templates") ARE intended to be reported in the instance.

(b) Instances MAY include appropriate negative (i.e. with @filed="false") filing indicator elements indicating reporting units which are intended NOT to be reported in the instance.

(c) Negative filing indicators MUST be included when a reporting unit3 is deliberately not reported which is potentially expected by the EBA to be contained in that instance (e.g. due to the reporter having no relevant transactions or positions to report, or on that occasion falling outside a relevant threshold for the reporting of the unit), in order to express the intention of the reporter not to report definite values for said template.[4]

1.6.1 — Multiple filing indicators for the same reporting unit

There is no benefit in filing several filing indicators for the same reporting unit. Inconsistent occurrences might occur (different values of @filed attribute).

Reported XBRL instances MUST contain only one filing indicator element for a given reporting unit ("template").

1.6.2 —Filing indicators in several tuples

Reporting filing indicator elements spread across several separate fIndicators tuples is a more complex approach than using a single containing element, and is likely to be more complex to handle by receivers.

However this construction may be useful for generating large instances (generation in a single pass or *streaming*), by allowing e.g. a tuple containing a single filing indicator to immediately precede (or follow) the data items for each reporting unit.

For flexibility, reported XBRL instances MAY include (different) filing indicators in several separate fIndicators tuple elements, for simplicity this SHOULD in general be avoided where not necessary.

[3] For which it is actually actual legitimate to not report definite values (this is not the case for all reporting units).

[4] CAs (and filers) should aim to comply with this requirement immediately; however the EBA does not intend to strictly enforce this rule earlier than the reporting cycle for reports with reference date Dec 2015 (i.e. no earlier than early 2016).

1.7 — Implication of no facts for an indicated template

If a positive filing indicator is given in the XBRL instance, appropriate consistency checks may be processed by the recipients' reporting system. If no facts appear for an indicated template, the filing may well be rejected because the system requires an appropriate, coherent set of fact values for the checks.

If there are no facts reported that match a template indicated with a positive filing indicator, this conveys that the template is intended to be explicitly reported and every numeric cell on that template may be considered (i.e. when applying validation checks) as equivalent to zero, not that the template as a whole is intended to be unreported[5]. In practice, this is unlikely to be the intent of a filer, and may indicate an error in instance preparation.

(a) Reported XBRL instances MUST include appropriate positive filing indicator elements to express which reporting units ("templates") are intended to be reported in the instance

(b) Reported XBRL instances MUST NOT include positive filing indicator elements indicating a reporting unit is filed (i.e. @filed=true) for reporting units which are NOT intended to be reported in the instance.

1.7.1 — No facts for non-indicated templates

Reported XBRL instances MUST NOT include facts which are not contained in any of the templates indicated by filing indicators as reported.

EBA Advice: Note that the combination of Rules 1.6 to 1.7.1 does NOT imply that there must be no facts in an instance which could be located on a template for which there is no positive filing indicator to indicate the template is reported. This IS possible in the specific situation that the fact is also included in a template which is indicated as reported (by a positive filing indicator) – i.e. where the same data point is present in multiple templates, at least one, but not all, of which are reported.

1.09 — Valid XML-XBRL

In order to increase the likelihood that instance documents pass validation, filers must validate their compliance with the XBRL 2.1 and Dimensional 1.0 specification prior to submission.

Instance documents MUST be XBRL 2.1 and XBRL Dimensions 1.0 valid. [EFM11, p. 6-8]

1.10 — Valid according to the defined business rules

XBRL allows the definition of business rules which can be discovered by XBRL software when opening the respective module referenced in the instance document. These business rules are applied on the content of the

[5] Which would be indicated with a *negative* filing indicator – and would indicate that any facts associated to the reporting unit (which are not anyway reported in the instance as part of another reporting unit with an associated positive filing indicator) are to be considered "unknown".

instance document to check the data quality. Tests that result in an error need to be corrected by the sending reporting entity.

(a) Instance documents MUST be valid with regards to validation rules as defined in the taxonomy (using XBRL formula), and discoverable from the referenced entry point, with the exception of any formula indicated as either deactivated or not mandatory to comply with in material published by the EBA.

(b) Instance documents MUST also be valid with regards to validation rules published in the applicable ITS, including those not implemented by the XBRL Formula, again with the exception of any validation rules marked as deactivated or non-mandatory in material published by the EBA.

```
context Instance::isValidationValid() : Boolean post: result = true
```

1.11 — Taxonomy extensions by reporters

XBRL Taxonomies can be extended by anybody with the proper technical knowledge. Filings to European Banking Authority are 'closed form' i.e. all data points allowed by the regulator are in the taxonomy. There can be no extension of the taxonomy by reporters to report more (or less) data points to the regulator. However national supervisors may extend European taxonomies. For reporters the combination of base and extension taxonomies is regarded as a single taxonomy.

Instances MUST reference only the taxonomy entry points specified by the relevant authority, and reporters MUST NOT provide their own extension taxonomies.

1.12 — Completeness of the instance

In case corrections are needed on filings that already have been sent, it is required to resubmit the complete filing, rather than partial data with just the corrected facts. Non-complete submissions could lead to invalid instance documents (according to either XBRL 2.1, XDT 1.0 or appropriate Formulae), might raise conflicts with already processed data in the reporting system of the receiver, and may lead to significant errors if sender and receiver disagree as to the list and sequence of historical submissions.

In the event of an amendment being required, instances MUST contain the full report – no content/values from previous instances may be assumed.

2. Instance syntax rules

2.1 — @xml:base

XBRL processors interpret this attribute differently, and there is no semantic need for this attribute.

XML-XBRL: The attribute xml:base may be inserted in XML documents to specify a base URI other than the base URI of the document or external entity.

The attribute @xml:base MUST NOT appear in any instance document. [EFM13, p. 6-7]

```
context xmlDocument inv:
self.element->select(xml:base)->isEmpty()
```

2.2 — xbrli:xbrl/link:schemaRef content

The taxonomy which is used by an XBRL report is identified by the URL(s) referenced by link:schemaRef elements. Although it is often convenient to work with local copies of the relevant taxonomies, it is important that link:schemaRef elements resolve to the published entrypoint locations. XBRL software typically provides functionality to “remap” references to URLs of published entrypoints to local copies of the taxonomy.

The link:schemaRef element in submitted instances MUST resolve to the full published entry point URL.

2.3 — xbrli:xbrl/link:schemaRef

Under the XBRL standard, the element link:schemaRef can occur several times in an instance. In the EBA taxonomy however only a single entry point schema needs to be referred to in any instance. This entry point will specify all required data points, and is the definition of a particular report.

Any reported XBRL instance document MUST contain only one xbrli:xbrl/link:schemaRef node.

```
context Instance inv: self.SchemaReference->size() = 1
```

2.4 — xbrli:xbrl/link:linkbaseRef

Entrypoints will be defined by means of a schema. There is no use for link:linkbaseRef.

Reference from an instance to the taxonomy MUST only be by means of the link:schemaRef node.

2.5 — XML comment and documentation

Comments inside the instance that do not get reported as a fact will be ignored by the EBA.

Relevant data MUST only be contained in contexts, units, schemaRefs and facts.

Context related rules

2.6 — xbrli:xbrl/xbrli:context/@id

The id attribute is meant as a unique technical key within a XML document. Conveying semantics in the id attribute will likely be lost when the XML content is processed, e.g. stored in a database (which generally works with database specific surrogate keys), any semantics are unlikely to be available to a (human) consumer of the instance data. Even though there is no limitation on the length of an id attribute it is recommended to keep it as short as possible.

Semantics SHOULD NOT be expressed in the xbrli:context/@id node.

2.7 — Unused xbrli:xbrl/xbrli:context

Unused contexts (contexts which are not referred to by facts) clutter the instance and add no value to either regulator or reporter [GFM11, p. 12].

Unused xbrli:context nodes SHOULD NOT be present in the instance. [FRIS04]

```
context Context inv: self.Fact.allInstances()->notEmpty()
```

2.8 — Identification of the reporting entity

The xbrli:identifier node combined with the @scheme allows the identification of the reporting entity by the receiver. The @scheme provides a URI which uniquely identifies the type of identifier used in the xbrli:identifier node (see section 3.6 LEI and other entity codes below).

(a) Instances MUST use a scheme that is prescribed by the by the receiving regulator. [GFM11, p. 11]

(b) Instances MUST use an identifier acceptable to the receiving regulator (likely to be one recognized in their reporting system), and that corresponds to the @scheme attribute used. [GFM11, p. 11]

(c) For remittance of data by competent authorities to the EBA, the entity identifier used should be a (pre)Legal Entity Identifier code, and must have been registered with the EBA by the CA prior to remittance.

2.9 — One reporter

There can only be one reporter of an instance. Even if the content of the instance deals with a group of companies, there is only one entity reporting the instance to the regulator.

All xbrli:identifier content and @scheme attributes in an instance MUST be identical. [EFM13, p. 6-8]

```
context Context inv: self.Identifier.allInstances()->forAll(i1, i2|
i1 = i2 implies i1.value = i2.value)
```

2.10 — xbrli:xbrl/xbrli:context/xbrli:period/*

The xbrli:startDate, xbrli:endDate and xbrli:instant elements all have data type which is a union of the xs:date and xs:dateTime types. European regulators will only allow periods to be identified using whole days, specified without a timezone.

All xbrli:period date elements MUST be valid against the xs:date data type, and reported without a timezone. [GFM11, p. 16]

2.11 — xbrli:xbrl/xbrli:context/xbrli:period/xbrli:forever

The extreme version of duration is 'forever'. The XBRL specification has created this to solve problems with dates starting 'at the beginning' and ending 'never'. E.g. the name of the filer has in general no end date. The EBA is only interested in data for the reported time segment, that has a defined starting and ending date.

The period 'xbrli:forever' MUST NOT be used. [GFM11, p. 19]

```
context Context inv: self.Period.forever->isEmpty()
```

2.13 — XBRL period consistency

XBRL requires all facts to be associated with a "period" (either a duration or instant of time). Where there are multiple relevant date/period like concepts related to a fact (as is often the case), it may be unclear which of these concepts is expressed by the XBRL period.

A common approach is to associate the XBRL period with some variation of a "real-world date of the event" for a fact. Use of varying "event" dates for facts in a regulatory reporting instance may however lead to complexity, confusion, and practical difficulties (e.g. for selecting facts for table linkbase axes, validating dates, identifying related facts etc.), particularly where the relationship between reporting periods and current and prior conceptual dates (e.g. accounting periods) is unclear, complex, and/or time-varying, such as in jurisdictions allowing non-calendar financial periods.

For simplicity therefore, the European Banking Authority has instead chosen to associate the "reference date" of an instance with the XBRL period concept.

The flow vs stock nature of a fact can be determined via a naming convention for the primary item, if the second character of the primary item name is "i" the fact is a stock (point in time measure), if "d" it is a flow or change.

Logical distinctions between other date-like aspects of a fact, such as the "event date, "applicable period", "date offset from reporting date" are conveyed via dimensional attributes of a fact.

All xbrl periods in a report instance MUST refer to the (same) reference date instant. All xbrl periods MUST be instants.

```
context Context inv: self.Period.allInstances()->forAll(p1, p2| p1 =
   p2 implies
p1.instant = p2.instant)
```

2.14 — xbrli:xbrl/xbrli:context/xbrli:entity/xbrli:segment and xbrli:xbrl/xbrli:context/xbrli:scenario

The XBRL Dimensions specification allows taxonomies to specify dimensions for use within either the segment or the scenario of the context. For consistency reasons and simplification of processing, the European Banking Authority only uses the xbrli:scenario node.

> **xbrli:segment elements MUST NOT be used.**

2.15 — xbrli:xbrl/xbrli:context/xbrli:entity/xbrli:segment and xbrli:xbrl/xbrli:context/xbrli:scenario

The xbrli:scenario or xbrli:segment element MUST NOT be used for anything other than for explicit or typed members. Custom reporter XML schema content may create problems with the regulatory system.

XML-XBRL: The XBRL specification allows xs:any content. This means that all XML schema content can be stored (not just XBRL Dimensions).

> **If an xbrli:scenario (or xbrli:segment) element appears in a xbrli:context, then its children MUST only be one or more xbrldi:explicitMember and/or xbrldi:typedMember elements, and MUST NOT contain any other (reporter custom) content. [EFM13, p. 6-8]**

Fact related rules

2.16 — Duplicate facts

An instance document must not have duplicated fact items. Item X and item Y are duplicates if all the following conditions apply:
1. X is not identical to Y, and
2. the element local name of X is S-Equal to the element local name of Y, and
3. X and Y are defined in the same namespace, and
4. X is P-Equal to Y, and
5. X is C-Equal to Y, and
6. X is U-Equal to Y, and
7. X and Y are dimensionally equivalent (d-equal in all dimensions of each of X and Y), and
8. (If X and Y are strings) X and Y have S-Equal xml:lang attributes

XML-XBRL: Duplicate facts are XML-XBRL syntax valid. However, the semantic meaning may be unclear.

> **Instances MUST NOT contain duplicate business facts. [FRIS04],[EFM13, p. 6-10]**

2.17 — @precision

The XBRL standard provides two methods of communicating the precision of a numeric fact: @precision and @decimals attributes. Humans seem to have an easier time reading a document that uses the decimals attribute, probably because the decimals value is likely to be only one of e.g. 2, 0, -3, -6, -9 or INF. Moreover, given a decimals value the precision can always be computed, but this is not symmetric.

> **@decimals MUST be used as the only means for expressing precision on a fact. [FRIS 2.8.1.1, EFM13, p. 6-12]**

2.18 — @decimals

The @decimals attribute indicates the accuracy of the reported fact value. If a numeric fact has an @decimals attribute with the value n then it is known to be “correct to n decimal places”.

The EBA will interpret this attribute on reported data as specifying that the absolute difference between the true value of the number as known to the reporter and its representation (known as the “absolute error” of the representation - e_{abs}) is less than or equal to 0.5 x 10^{-n}, and reporters must prepare submitted reports consistently with this interpretation[6].

The EBA XBRL validation rules use interval arithmetic for validation. To enable XBRL Formula calculations to be best performed on instance values for validation purposes, no truncations or rounding or any other kind of change should apply to the numeric facts in the instance. See also the explanatory RFC at http://www.xbrl.org/RFC/PDU/PWD-2008-10-09/PDU-RFC-PWD-2008-10-09.html

(a) The accuracy of a numeric fact MUST be expressed using @decimals

[6] See also the explanation of “Correct to n decimal places” given in the (now superseded) 2008-07-02 Errata version of the XBRL 2.1 specification at http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm#_4.6.7.2

(b) There SHOULD be no truncation, rounding or any change in the original fact value, which should be reported as known.

EBA Note: In particular, if numbers are truncated or rounded for reporting, they should not be "adjusted" so that they "appear" to be visually consistent (i.e. so that they "foot" or "cast"), but should instead be simply reported with the appropriate @decimals value – the validation checks will take into account the declared accuracy to determine if reported values are (could be) valid.

Accuracy Requirements			
Data Type	**Decimals attribute**	**Note**	**Representation**
Monetary[7]	>= -3		42563.26
Percentage	>= 4	Must be expressed as a ratio in instances – i.e. typical values between 0 and 1	0.1234 (=12.34%)
Integer	0	Must of course be reported without any decimal part	126

N.B. INF (meaning exact as written) is of course acceptable for the decimal attribute of all numeric types.

EBA Note: This, combined with the definition of the @decimals property, means that monetary values must not be truncated *to thousands (since the reported value might then be up to 1000 from the true value, which is more than the 500 implied by @decimals=-3, requiring instead decimals=-4 to be consistent), but may be* rounded *to thousands.*

The decimals attribute is not a scale factor. The decimals attribute is not a formatting code; it does not indicate that the digits in the instance must subsequently be presented to a user in any particular way.

The @decimals attribute influences how numbers are interpreted. Use the following table to select the correct value of the @decimals attribute for a fact so that it corresponds to the accuracy to which the value is known.

Accuracy of the amount	Value of decimals attribute
Exact monetary, percentage, basis point or any other	INF
Accurate to thousands	-3
Accurate to hundreds	-2
Accurate to units	0
Accurate to cents	2
Accurate to a hundredth of a percentage point (i.e. a *basis point*)	4

[7] N.B. Also applies to facts representing monetary values that are specified (via their primary item) to be reported as currency-less decimal values.

Examples: The table below illustrates correct use.

Data	Reported Value	Value of @decimals attribute	Range of value considered in interval arithmetic
A percentage (ratio) of (exactly) 46%	0.46	INF	0.46
A profit margin of 9.3% (minimum accuracy)	0.093	4	0.09295 to 0.09305
Monetary amount “in thousands”	100000	-3	99500 to 100500
Monetary amount “in hundreds”	100200	-2	100150 to 100250
Monetary amount, accuracy of “units”	100205.23	0	100204.73 to 100205.73

[EFM13, p. 6-28], [GFM11, p. 45f.]

EBA NOTE: For clarification - this guidance applies only to the representation of the values in the transmission XBRL instance file, it of course places no constraints on the display of information to any user or preparer of the data. Tools may choose to display values however they (and their user’s) desire, so long as when instance files are produced the canonical representation is used.

2.19 — zero value, empty, nil value @xsi:nil

Data related to white cells could be reported with a non-zero value, reported as zero or unreported. The table below shows the different possible scenarios:

Reported Zero or Non-zero value	e.g. <eba_met:mi53 unitRef="uEUR" decimals="2" contextRef="c2">1025.25</eba_met:mi53>		The value of the fact is known.
Reported nil value	e.g. <eba_met:mi53 unitRef="uEUR" decimals="2" contextRef="c2" @xsi:nil="true" />		MUST NOT be used
Missing fact	The fact doesn't appear in the instance.	Template including this fact is reported *(i.e. positive filing indicator included)*	The value is treatable as equivalent to zero (if numeric) by the recipient.
		No template including this fact is reported *(i.e. negative filing indicator included, or indicator absent)*	The value is “unknown” to the recipient.

Inapplicable information need not be included in an instance, i.e. inapplicable facts MAY be left out.

EBA Note: For validation purposes, unreported numeric facts belonging to a template indicated as "reported" by an instance (using filing indicators) will be treated as equivalent to zero in the evaluation of certain rules – see the details of individual rules.

EBA Note: Zero values SHOULD, preferably, be explicitly reported where they are interesting supervisory reporting information. "Uninteresting zeros" (i.e. large swathes/permutations of trivially zero or simply inapplicable information, for example the large bulk of countries, currencies, lines of activity etc. in which a reporter has nothing relevant to report) SHOULD NOT be reported for obvious practical reasons.

2.20 — @xml:lang

The language used on string based facts may need to be identified. This can be done by declaring the @xml:lang on the xbrli:xbrl element just once, or on every string based fact individually. No restrictions are places on language used in reporting string facts (such as entity names), however some strings are required to have specific values by the ITS which are not language specific, and should be the same whatever language is marked. In practice, the @xml:lang attribute is in general not required in instances remitted to the EBA and may be omitted.

> **An @xml:lang attribute MUST be used when required to distinguish otherwise duplicate string facts, where an individual fact is reported in more than one language (i.e. with translation).**

This is expected to be a relatively rare situation as there is no requirement to submit translations of string facts.

Unit related rules

2.21 — Duplicates of xbrli:xbrl/xbrli:unit

Units are equivalent if they have equivalent measures or equivalent numerator and denominator. Measures are equivalent if their contents are equivalent QNames. Numerators and Denominators are equivalent if they have a set of equivalent measures. Duplicated units do not express extra semantics and potentially disturb comparison of facts that point to any of the duplicated occurrences [EFM13, p. 6-10].

> **An XBRL instance SHOULD NOT, in general, contain duplicated units, unless required for technical reasons, e.g. to support XBRL streaming.**

2.22 — Unused xbrli:xbrl/xbrli:unit

Unused units (units which are not referred to by facts) clutter the instance and add no value to either regulator or reporter.

> **An XBRL instance SHOULD NOT contain unused xbrli:unit nodes. [FRIS04]**

2.23 — xbrli:xbrl/xbrli:unit/* content

XII has released a standard numeric data type registry: it has a schema with numeric type declarations, and each numeric data type is associated with consistent unit declaration measures, numerators and denominators. Use of this registry that contains all the usual units eases implementation in software and simplifies validation (http://www.xbrl.org/utr/utr.xml).

> **xbrli:unit children SHOULD refer to the XBRL International Unit Type Registry (UTR). [EFM13, p. 6-17]**

2.24 — xbrli:xbrl/xbrli:unit/xbrli:measure

Facts that represent amounts in any currency will generally be of an item that is derived from xbrli:monetaryItemType, which must follow the restriction in XBRL 2.1, section 4.8.2, regarding monetaryItemType (i.e., unit measure is an ISO 4217 currency designation). Such facts must not have unit measures that express any scaling (which would interfere with the expression of accuracy by the @decimals attribute).

> **Units representing currencies MUST represent the actual physical value of these currencies, i.e. in basic units, not including any scaling factor in the unit.**

3. Additional Guidance

3.1 - One Explicit Currency

An instance MUST express its monetary facts[8] using a single currency.[9]

3.2 - Non-monetary numeric units

(a) An instance MUST express its non-monetary numeric values using the “pure” unit, a unit element with a single measure element as its only child. The local part of the measure MUST be "pure" and the namespace prefix MUST resolve to the namespace: http://www.xbrl.org/2003/instance .

(b) Rates, percentages and ratios MUST be reported using decimal notation rather than in percentages where the value has been multiplied by 100 (e.g. 9.31% must be reported as 0.0931).

3.3 - Decimal representation

The value of numeric facts MUST be expressed in the specified units, without any change of scale and SHOULD ideally be expressed without rounding or truncation.

The content of a numeric fact MUST therefore not include any scale factors like “%”. Specifically, monetary values[10] MUST be expressed in units, not in thousands or millions.

i.e. the value €2,560,561.43 may be transmitted as, amongst others, any of

Acceptable representations of €2,560,561.43		
Value	**Value of decimals**	**Implies**
2560561.43	INF	Exact
2560561.43	2	+/- 0.005
2560561.43	0	+/- 0.5
2560561.43	-3	+/- 500
2560561	*0*	*+/- 0.5*
2561000	*-3*	*+/- 500*

[8] i.e. items of monetaryItemType. N.B. this rule does NOT apply to facts representing monetary positions that are explicitly indicated by the data type of the primary item as being required to be reported as “currency-less” decimal values (the value for which may be required to be based on a currency that is not the main currency of the report).

[9] For clarity – where providing a breakdown by currency, the value of an item in the non-reporting currency should be converted to the equivalent value in the reporting currency (e.g. 2USD -> 1.44 EUR) for submission (the data item being identified as corresponding to an exposure in the breakdown currency by its dimensional attributes) . Again, this rule does not apply to facts representing monetary positions which are to be reported using metrics of a decimal data type – for these the specific instructions for the report should be followed as to whether conversion to the reporting currency is required.

[10] Whether using monetaryitemtype metrics or decimal.

Note that although the last two representations (rounding the transmitted value) are acceptable, EBA would prefer that they are avoided where a better estimate for the value is known, and this is transmitted without rounding or truncation as in the first four examples.

But, for example, €2,560,561.43 MUST NOT be transmitted as “2561”

Unacceptable representation of €2,560,561.43	
Value	Value of decimals
2561	-3

As this represents €2,561 (+/-500), rather than the intended €2,561,000.00 (+/-500)

3.4 Unused namespace prefixes

Declaring unused namespaces is uncalled for and clutters the instance document.

> **Namespace prefixes that are not use SHOULD not be declared in the instance document. [FRIS04]**

3.5 Re-use of canonical namespace prefixes

Most schema authors provide a namespace prefix for their targetNamespace. It is common practice to re-use these prefixes in other XML documents when needed. It may lead to confusion to human readers to see common understood prefixes used on a different namespace. E.g. prefix 'xs' for the http://xbrl.org/2003/xbrl-instance-2033-12-31 namespace.

> **Namespace prefixes declared in instance documents SHOULD mirror the namespace prefixes as defined by their schema author(s). [FRIS04]**

3.6 LEI and other entity codes

Practical Considerations

For second level remittance to the EBA, the entity code used MUST be pre-registered with the EBA by the appropriate CA.

Guidance on representation of codes as entity identifier

LEIs

The EBA requires the use of “http://standard.iso.org/iso/17442” as the scheme identifier for pre-LEIs, i.e.

```
<xbrli:entity>
  <xbrli:identifier
scheme="http://standards.iso.org/iso/17442">LEIIDENTIFIERABCDEFG</xbrli:identifier>
</xbrli:entity>
```

where LEIIDENTIFIERABCDEFG is replaced with the appropriate pre-LEI code for the entity.

Other Identifiers

In general, i.e. for first level remittance, or for specific data collections, the scheme URI (and entity code) to be used in an instance should be determined by the relevant competent authority.

4. Examples

Filing indicator examples

Consider a report containing information for templates X1, and X2, but not X3. The typical approach to indicating this with filing indicator elements would be:

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">X1</find:filingIndicator>
  <find:filingIndicator contextRef="c1">X2</find:filingIndicator>
</find:fIndicators>
…some data…
```

Here there is a single “fIndicators” element grouping two filing indicator elements, which indicate the intention to report X1 and X2.

Some **acceptable variations** of this include using the filed attribute:

```
<find:fIndicators>
  <find:filingIndicator contextRef="c2">X1</find:filingIndicator>
  <find:filingIndicator contextRef="c2" find:filed=”true”>X2</find:filingIndicator>
</find:fIndicators>
…some data…
```

Or utilising more than one containing “fIndicators” element:

```
<find:fIndicators>
  <find:filingIndicator contextRef="A" find:filed=”true”>X1</find:filingIndicator>
</find:fIndicators>
…some data…
<find:fIndicators>
  <find:filingIndicator contextRef="A">X2</find:filingIndicator>
</find:fIndicators>
…some more data…
```

It is also acceptable (and in some cases required) to explicitly indicate that the X3 template is NOT reported, e.g.

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">X1</find:filingIndicator>
  <find:filingIndicator contextRef="c1">X2</find:filingIndicator>
  <find:filingIndicator contextRef="c1" find:filed=”false”>X3</find:filingIndicator>
</find:fIndicators>
…some data…
```

Unacceptable variations include, for example:

Not indicating that a reported template is reported (X2 is missing):

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">X1</find:filingIndicator>
</find:fIndicators>
…some data…
```

Indicating that an unreported template is reported (X3 is not reported):

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">X1</find:filingIndicator>
  <find:filingIndicator contextRef="c1">X2</find:filingIndicator>
  <find:filingIndicator contextRef="c1">X3</find:filingIndicator>
</find:fIndicators>
…some data…
```

Duplicating a filing indicator (here both X1 and X2 appear twice, either repetition is an error):

```
<find:fIndicators>
  <find:filingIndicator contextRef="A" find:filed=”true”>X1</find:filingIndicator>
  <find:filingIndicator contextRef="A">X1</find:filingIndicator>
  <find:filingIndicator contextRef="A">X2</find:filingIndicator>
</find:fIndicators>
…some data…
<find:fIndicators>
  <find:filingIndicator contextRef="A">X2</find:filingIndicator>
</find:fIndicators>
…some more data…
```

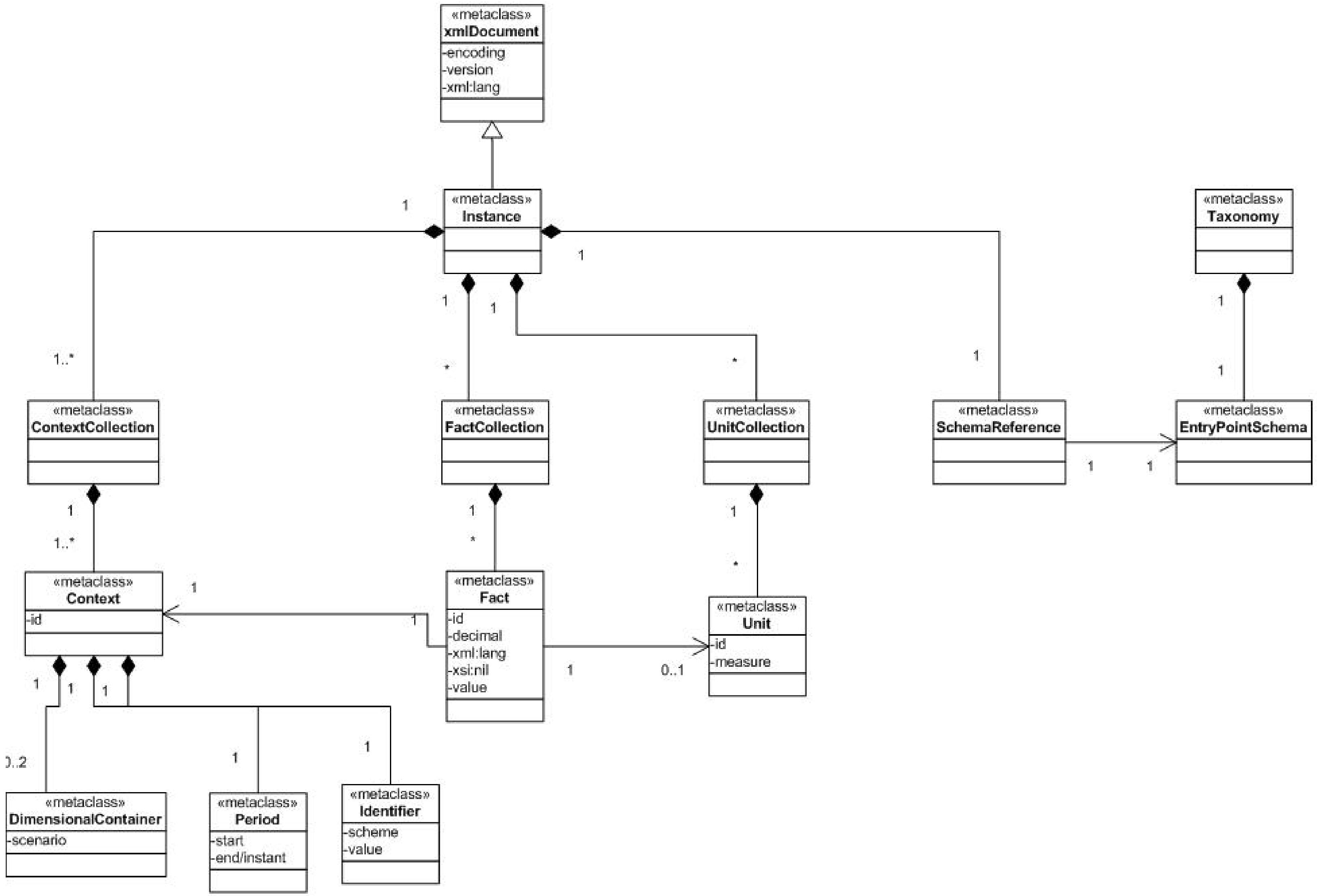
Consider also a template X4, known to be expected/anticipated by the recipient (i.e. EBA) to be reported in this instance by this filer. In this case, if this template is not in fact reported, it is it is not acceptable to omit to indicate this explicitly:

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">X1</find:filingIndicator>
</find:fIndicators>
```

Should instead be:

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">X1</find:filingIndicator>
  <find:filingIndicator contextRef="c1" find:filed=”false”>X4</find:filingIndicator>
</find:fIndicators>
```

European Filing Rules: UML model


«metaclass»
xmlDocument
-encoding
-version
-xml:lang
«metaclass»
Instance
«metaclass»
Taxonomy
1
1
1
1
1..*
*
*
1
1
1
«metaclass»
ContextCollection
«metaclass»
FactCollection
«metaclass»
UnitCollection
«metaclass»
SchemaReference
«metaclass»
EntryPointSchema
1
1
1
1
1
1..*
*
*
«metaclass»
Context
-id
1
1
«metaclass»
Fact
-id
-decimal
-xml:lang
-xsi:nil
-value
1
0..1
«metaclass»
Unit
-id
-measure
1
1
1
).2
1
1
«metaclass»
DimensionalContainer
-scenario
«metaclass»
Period
-start
-end/instant
«metaclass»
Identifier
-scheme
-value