EBA/XBRL/2021/10

13 October 2021

Filing Rules version 5.1

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 templates that are "expected" to be reported (i.e. for which there is a high likelihood a reporter will need to submit the report, most banks of that kind are expected to have events reported in these 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
3.1	April 2015	Correction of the scheme identifier for LEIs and pre-LEIs

4	June 2015	Rules 1.13 to 1.15 were added. Rules 1.6, 2.16.1 and 2.18 have been updated with significant changes. More guidance is provided by adding rules 2.25 and 3.7 to 3.10. Minor changes have been done for clarification and better understanding. In the section Terms and definitions a new term “byte order mark” was added and the definitions for “fact” and “filing indicators” were improved. With the establishment of the SSM the wording in the section “Scope of application” has undergone some changes. For the “Filing syntax rules” there were changes applied to rule 1.6. A new sublevel rule 1.6 (d) was incorporated to add a constraint to the declaration of filing indicators in the instance document. Moreover a table was added to clarify the use of the @find:filed attribute for filing indicators for remittance to the EBA. Rule 1.6.3 was reworded to clarify that only valid filing indicators may be used respective to the reported instance. The EBA note for rule 1.7.1 was rewritten for clarification. The new text for rule 1.12 emphasizes that not only resubmissions but also the first submission of a reporter must be complete. In the section “instance syntax rules” a new rule 2.25 has been added to include information about the use of XBRL footnotes in instances. The rephrasing of rule 2.16 together with the improved definition of facts compared to business facts improves the explanation of the occurrence of duplicates. Furthermore the captions in this section have been reworded to be more self-describing. Rule 2.16.1 has been rewritten for clarification. A new sublevel rule 2.18 (c) was added to emphasize that the @decimals attribute used should be realistic. Also a new row was included in the table provided with this rule to indicate the accuracy of millions allowed for the module Funding Plans. The rules 3.1 to 3.3 were moved to the section “instance syntax rules”. The section “additional guidance” was extended by four additional rules that should be noted. The filing rule examples at the end of the document were adjusted with more concrete examples and clearer formatting. Moreover examples for the new guidance on namespace prefix declaration were included. Furthermore the file naming structure for remittance to the EBA was added. Section highlighting the impact of “streamable” instance preparation on the application of guidance rules was added. Reference to the CEN (European Normalization Centre) publication about European Filing Rules was updated.
4.1	August 2015	Adapted to allow multicurrency reporting as per EBA Single Rulebook QA #1042 - change to rule 3.1 (pp37-38) Correction to LEI URI (“correction” in v3.1 sadly changed the one of the two variations used that was in fact correct into the incorrect form)
4.2	November 2016	Added tags/names for various filing rules to aid identification etc. Added further explanatory material on multi-currency instances to the examples section.
4.3		Minor changes. Slight clarification on usage of LEI / other possible reporting subject coding by the EBA (see 2.8, 3.6 etc). Guidance on avoidance of potentially misleading whitespace in strings. Request for use of xml processing instructions to convey information about the software component used for production of the XBRL instance.
4.4		Specify how to report the liquidity subgroup reporting documents in 3.6. The file naming structure for remittance to the EBA is amended to specify the report subject for the liquidity subgroup reporting documents and to specify that only the zip format file will be accepted These new requirements will be applied from 2021-01-01; these new requirements will also be applied to all the resubmissions from 2021-01-01.

4.5		Rule 2.19 is updated by specifying that empty string is not allowed to be reported for the string type metric
5.0	May 2021	The particularities around the usage of the both xBRL-XML and xBRL-CSV variants are included for each relevant section. The term "(XBRL) instance (file)" is replaced by "(XBRL) report", which occurs in various linguistic combinations.
	July 2021	Modify 1.6.a to remove the possible misinterpretations and add a new rule 1.6.e to clarify that a reporting document must contain at least one filing indicator (positive or negative)
5.1	October 2021	Modify rule 1.6 to simplify the usage of filing indicator by removing the option "empty filing indicator". This updated rule must be followed from 2023-01-01 for all the submissions and resubmissions. Modify rule 3.6 to improve CON/IND information circulation and to allow for multiple consolidation levels for an entity, CON/IND information is being moved from module name to reporting subjects from reference date 31/12/2022 onwards.

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
xBRL-CSV: mapping from Open Information Model 1.0
CWA European Filing Rules

Bibliography

[CWA] CEN Workshop Agreement 16744-4:2014 Improving transparency in financial and business reporting - Harmonisation topics - Part 4: European Filing Rules (ftp://ftp.cen.eu/CWA/CEN/XBRL/CWA_16744-4_2014.pdf)

[EBA14] Representation in XBRL of the Data Point Model (http://www.eba.europa.eu/documents/10180/632822/EBA+Architecture+for+XBRL+representation+of+DPM.pdf)

[EFM13] EDGAR Filer Manual. U.S. Securities and Exchange Commission (http://www.sec.gov/info/edgar/formdxmltechspec.htm)

[GFM11] Global Filing Manual (Interoperable Taxonomy Architecture Project)
(http://www.ifrs.org/XBRL/Resources/Documents/GlobalFilingManual20110419.pdf)

[IEEE754] IEEE Standard for Floating Point Arithmetic, IEEE Std 754-2008
(http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4610933)

[RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
(http://www.ietf.org/rfc/rfc2119.txt)

[SBR13] SBR FRIS rules 2013
(http://www.sbr-nl.nl/fileadmin/SBR/documenten/NT_2013/definitief_03122012/NL-FRIS_NT2013_20121210.pdf)

[EIOPA15] EIOPA XBRL Filing Rules for Solvency II reporting
(https://dev.eiopa.europa.eu/Taxonomy/Full/2.2.0/EIOPA_XBRL_Filing_Rules_for_Solvency_II_reporting_2.2.0.pdf)

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 (http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html)

applicable taxonomy

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

byte order mark

In UTF-8 documents, a sequence of characters (0xEF, 0xBB, 0xEF) that may be used to signal that the characters' are encoded using UTF-8 but, in this particular case, its use is neither required nor recommended by the Unicode consortium

competent authority (CA)

legally responsible authority

data point

a Data Point is an information component that is defined by a supervisor to be sent in an XBRL report

Note: In XBRL a data point is represented by a fact 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

entry point

the starting point for discovery of the filing requirements referenced from an XBRL report.
Entry points come in two flavours: as XML schema (XSD) and as a JSON file.
Where required, a specific term is used to identify which technology is referred to. The terms used are "XSD entry point" for XML Schema and "JSON entry point" for the JSON file.
Where the general term "entry point" is used, both ways are referred to.

fact

a fact is a value reported against a datapoint in an XBRL report

A business fact is a fact that conveys a business value. Filing indicators facts are not business facts

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 report or series of XBRL reports

filing indicators

indicate the reporting units (typically templates) reported in the report

Note: Filing indicators are facts, according to XBRL definitions, but they have special characteristics and are not subjects to the rules defined in this document which cover all other type of facts, called business facts

filing system

a system in which XBRL reports are filed, received, checked, stored, analysed and redistributed

json entry point

see entry point.

reporter

a reporting entity – described by report(s)

reporting unit

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

template

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

XSD entry point

see entry point.

Introduction

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

Scope of Application

The European supervisory reporting process is conceptually a multi-stage process, first institutions prepare, validate and remit supervisory data to their relevant national authorities ("first level reporting"), where applicable, some data are sent to a supranational authority, and subsequently those authorities remit data to the European Banking Authority ("second level reporting").

These filing rules represent a collection of additional rules and guidance specifically **applicable to the remittance** of XBRL reports for reporting entities in scope of relevant EBA regulations (e.g. banks) regulatory filings **by relevant national and supranational authorities to the European Banking Authority**.

Focussed on the preparation of XBRL reports, rather than details of the mechanics of report submission/data collection these rules constrain the full flexibility of XBRL, to enable effective interaction between transmitter and recipient/consumer of regulatory filings.

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

This document was reviewed by a group of national experts in order to clarify any misleading formulation of rules and contribute to the pan-European harmonisation of the filing rules. The rules as stated in this document are those enforced on the second level of reporting (to EBA). In the case of supervisory authorities adopting these rules but with adaptations, for example changing preferences or guidance expressed by the EBA instead into obligations on the first level of reporting such variations will be communicated to the reporter by the respective supervisor.

Note: these rules are not necessarily those that are applicable at the level of reporting by individual institutions or groups of institutions, nor do they address the entire scope of the reporting process. Guidance should be sought from the reporter's competent authority as to their reporting format and requirements for that reporting.

Note also: by their nature, not all of these rules will be possible/ practical to determine, implement and enforce in an automatic manner, and in several cases simply declare or explain expected practice on behalf of reporters.

Basis in harmonized "European Filing Rules" guidance

In order to promote and enhance interoperability, these rules are largely drawn from the document CWA 16744-4:2014 European Filing Rules, promulgated by the European Normalization Centre (CEN), which *"represent a collection of recommendations to be seen as guidance to be implemented in the European supervisory reporting process"*. This document should be read in conjunction/comparison with that CEN document.

Numbering of rules

Please note that the rules are not necessarily numbered in sequential order. For ease of comparison, rules were originally numbered as per their numbering in the CEN document hence some numbers were omitted where the corresponding CEN rule was not applicable/not included. To aid identification and comparison between revisions of this document where possible the initial numbering of specific rules is retained, hence rules may be out of order, or in different sections from that implied by their numbering.

Many rules have been given specific identifying tags or names, e.g. "DuplicateFact". This is in order to aid identification.

Target Audience

Although primarily addressed to those (mostly technical staff) within the national and supranational authorities responsible for preparation or submission of XBRL reports directly to the European Banking Authority, these filing rules will also be of value to individual reporters (i.e. financial institutions or groups of institutions) reporting to those authorities which may utilise the EBA filing rules or XBRL format, or derivatives of them.

This document is intended for a technical audience and assumes that the reader has a working knowledge of the XBRL 2.1 specification, and other specifications such as XBRL Dimensions 1.0 and XBRL Open Information Model 1.0, alongside 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.

Relationship to Other Work

This document should be read in conjunction with the EBA Taxonomy Architecture. [EBA14]

The guidelines in this document pertain to XBRL reports. 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/).

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.

.

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

Use of Language

The use of language in this document follows that specified in [RFC 2119], in summary:

The use of “MUST” implies an obligation, and the preparation of XBRL reports not following these rules will generally result in rejection of the report.

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 not result in rejection of an XBRL report by EBA.

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

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

About the filing rules structure.

Further in this document, filing rules specify constraints which apply in general to XBRL reports. If there is no mentioning of xBRL-CSV or xBRL-XML then the rule does not apply to that syntax.

1. Filing syntax rules

1.1 — Filing naming

xBRL-XML reports

Common practice is to use the extension .xbrl for xBRL-XML reports. Detailed file naming requirements should be confirmed with the intended recipient of an XBRL report. Credit institutions should confirm with their relevant supervisor for reporting. The file naming convention to be used by CAs for remittance to the EBA can be found in the examples section.

xBRL-CSV reports

Common practice is to group the set of files in a zip container. Detailed file naming requirements should be confirmed with the intended recipient of the XBRL report. Institutions should confirm with their relevant supervisor for reporting. The structure of this zip container and the file naming convention to be used by CAs for remittance to the EBA can be found in the examples section.

1.4 — Character encoding of XBRL reports

Rule

All XBRL reports must use the UTF-8 character encoding (regardless of with or without BOM) in order to ensure that the receiver is able to process it.

Implementation for xBRL-XML and xBRL-CSV reports

Note that, as per https://www.w3.org/TR/xml/#charencoding, character encoding names should be matched in a case-insensitive way, so UTF-8 and utf-8 are equally acceptable.

encodingNotUtf8: XBRL reports MUST use UTF-8 encoding. [GFM11, p. 11]

1.5 — Taxonomy entry point selection

Rule

A taxonomy is loaded through a reference to one or more URLs. 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.

Implementation for xBRL-XML reports

The selection of the specific module when using xBRL-XML is done through the schemaRef element. This schemaRef must contain the XML schema defined by the EBA for that module. The taxonomy also contains other XML schemas, these are not to be used as xsd entry points.

a) multipleSchemaRefs: **Reporting entities MUST reference only one xsd entry point ("module", link:schemaRef element), as specified in the applicable taxonomy, per XBRL report. [SBR13, p. 6]**

b) inappropriateSchemaRef: **The schemaRef element MUST refer to a URL appropriate to the module and the reference date of an XBRL report, drawn from the list of xsd entry points published by the EBA[1].[EBA14]**

Implementation for xBRL-CSV reports

The selection of the specific module when using xBRL-CSV is done through the "extends" element. This element must contain the JSON entry point file defined by the EBA for that module. The taxonomy also contains other JSON files, these are not to be treated as entry points.

(a) multipleTaxonomyRefs**: Reporting entities MUST NOT populate the element documentInfo \ taxonomy. The reference to the taxonomy is done through the json entry point ("documentInfo", "extends" element).**

(b) inappropriateTaxonomyRef**: The documentInfo \ extends element MUST contain a single reference to the URL appropriate to the module and the reference date of a report, drawn from the list of json entry points published by the EBA[1].[EBA14]**

1.6 — Filing indicators

Rule

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

A filing indicator element 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 usage 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.

Implementation for xBRL-XML reports

The filing indicator element is called filingIndicator and is grouped (potentially with other such elements) within a containing element fIndicators.

(a) missingPositiveFilingIndicator**: XBRL reports MUST include appropriate positive (i.e. either with @find:filed="true" or without @find:filed attribute) filing indicator elements to express which reporting units ("templates") ARE intended to be reported. [2]**

(b) missingNegativeFilingIndicator**: XBRL reports MUST include appropriate negative (i.e. with**

[1] or competent authority for first level reporting.

[2] This rule does not require that at least one positive filing indicator must exist, it simply requires to have positive filing indicators for intentionally reported templates.

@find:filed="false") filing indicator elements to express which reporting units ("templates") are intended NOT to be reported.

(c) invalidContextForFilingIndicator**: The context referenced by the filing indicator elements MUST NOT contain xbrli:segment or xbrli:scenario elements.**

(d) missingFilingIndicators: An XBRL report MUST provide a filing indicator (either positive or negative) for each reporting unit ("template") defined in the reporting module to which the XBRL reports refers.

Implementation for xBRL-CSV reports

The filing indicator element is named filed and is of type Boolean. The name of the template is provided through a typed dimension linked to the element.

(a) missingPositiveFilingIndicator**: XBRL reports MUST include appropriate positive (i.e. with a fact value of true) filing indicator elements to express which reporting units ("templates") ARE intended to be reported.**

(b) missingNegativeFilingIndicator: **XBRL reports MUST include appropriate negative (i.e. with a fact value of false) filing indicator elements to express which reporting units ("templates") are intended NOT to be reported.**

(d) missingFilingIndicators: An XBRL report MUST provide a filing indicator (either positive or negative) for each reporting unit ("template") defined in the reporting module to which the XBRL reports refers.

Selected example scenarios:

Scenario	Positive filing indicator	Causes rejection
A template is included in the reported report with facts	true	No
A template is included in the reported report, but no associated facts are *explicitly* reported (i.e. included in the XBRL report).	true	No (all facts for template may be assumed to be zero, see 1.7)
A template isn't reported in the report but exist in the reporting module	false	No
A template is not reported in the report but exist in the reporting module	n/a	Yes (violation of 1.6)
Fact values for a template are reported, at least some of which are not also part of another template which has a positive filing indicator	false	Yes (violation of rule 1.7.1)

Scenario	Positive filing indicator	Causes rejection
A template is not reported, but facts "appearing on that template" *are* reported, they are all contained in other template(s) which *are* indicated as reported in the report	false	No (see EBA advice to 1.7.1)
A template is reported. Multiple filing indicators with the same code are included in the report.	n/a	Yes (violation of rule 1.6.1)

1.6.1 — Multiple filing indicators for the same reporting unit

Rule

There is no benefit in filing several filing indicators for the same reporting unit. Inconsistent occurrences might occur.

Implementation for xBRL-XML and xBRL-CSV reports

duplicateFilingIndicator**: XBRL reports MUST contain only one filing indicator element for a given reporting unit ("template").**

1.6.2 —Filing indicators in several tuples

Rule

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 reports (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.

Implementation for xBRL-XML reports

filingIndicatorInMultipleTuples**: For flexibility, reported XBRL reports MAY include different filing indicators in several separate fIndicators tuple elements, for simplicity this SHOULD in general be avoided where not necessary.**

1.6.3 – Filing indicator codes

Rule

As stated in the EBA Taxonomy Architecture the values of filing indicators to be used are indicated by label resources associated with the tables in the XBRL taxonomy. The value used should be exactly as indicated.

Implementation for xBRL-XML and xBRL-CSV reports

invalidFilingIndicatorValue: **The values of filing indicators MUST only be those given by the label**

resources with the role http://www.eurofiling.info/xbrl/role/filing-indicator-code applied to the relevant tables in the XBRL taxonomy[3] for that reporting module (entry point). Filing indicator values must be formatted correctly (for example including any underscore characters).

1.7 — Implication of no facts for an indicated template

Rule

If a positive filing indicator is given in the XBRL report, 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 cell on that template may be considered (i.e. when applying validation checks) as equivalent to zero (for numeric value) or blank (for non-numeric), not that the template as a whole is intended to be unreported[4]. In practice, this is unlikely to be the intent of a filer, and may indicate an error in report preparation.

Implementation for xBRL-XML and xBRL-CSV reports

(a) missingPositiveFilingIndicator: **XBRL reports MUST include appropriate positive filing indicator elements to express which reporting units ("templates") are intended to be reported in the report**

(b) positiveFilingIndicatorForNonReportedUnit: **XBRL reports MUST NOT include positive filing indicator elements indicating a reporting unit is filed for reporting units which are NOT intended to be reported in the report.**

1.7.1 — No facts for non-indicated templates

Rule

All facts must be intentionally provided by the reporter.

Implementation for xBRL-XML and xBRL-CSV reports

reportedFactAssociatedWithNoPositiveFilingIndicator: **XBRL reports MUST NOT include business facts which are not contained in any of the reporting units ("templates") indicated by filing indicators as reported.**

EBA Advice: Note that a single fact may notionally appear in several reporting units ("templates") - i.e. cells from several templates may represent the same data item. It may be the case that only some of these templates are reported in a report, and others are not. In these situations the presence of such a fact which is part of a reported

[3] N.B. equivalent information is available in the EBA DPM Database.

[4] 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 XBRL report as part of another reporting unit with an associated positive filing indicator) are to be considered "unknown"

template but which would also be part of an unreported template is NOT a breach of these rules – i.e. they do not require that all templates containing an reported fact are indicated as reported, just that all reported facts appear in at least one template which is indicated as reported.

1. 9 — Valid XBRL

In order to increase the likelihood that XBRL reports pass validation, filers must validate their compliance with the relevant XBRL specifications prior to submission.

Implementation for xBRL-XML reports

notValidXbrlDocument: **xBRL-XML reports MUST be XBRL 2.1 and XBRL Dimensions 1.0 valid. [EFM11, p. 6-8]**

Implementation for xBRL-CSV reports

notValidXbrlDocument: **xBRL-CSV reports MUST be xBRL-CSV 1.0 valid.**

1.10 — Valid according to the defined business rules

XBRL allows the definition of business validation rules which can be discovered by XBRL software when opening the respective module referenced in the report document. These business validation rules are applied on the content of the report document to check the data quality.

Implementation for xBRL-XML and xBRL-CSV reports

(a) notValidAccordingToTaxonomyValidationRules: **XBRL reports MUST also be valid with regards to validation rules published in the applicable ITS, including those not implemented by the validation rules as defined in the taxonomy (using XBRL formula), again with the exception of any validation rules marked as deactivated or non-mandatory in material published by the EBA**

(b) notValidAccordingToITSValidationRules: **XBRL reports MUST also be valid with regards to validation rules published in the applicable ITS, including those not implemented by the validation rules as defined in the taxonomy (using XBRL formula), again with the exception of any validation rules marked as deactivated or non-mandatory in material published by the EBA.**

1.11 — Taxonomy extensions by reporters

Rule

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 supervisor are in the taxonomy. There can be no extension of the taxonomy by reporters to report more (or less) data points to the supervisor. However some CA's may extend European taxonomies. For reporters the combination of base and extension taxonomies is regarded as a single taxonomy (also see 1.5).

Implementation for xBRL-XML reports

inappropriateSchemaRef: **xBRL-XML reports MUST reference only a XSD entry point specified by the supervisor (i.e. reporters MUST NOT reference their own extension taxonomies).**

Implementation for xBRL-CSV reports

inappropriateTaxonomyRef: **xBRL-CSV reports MUST reference only a JSON entry point specified by the supervisor (i.e. reporters MUST NOT reference their own extension taxonomies).**

1.12 — Completeness of the report

Rule

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 XBRL report (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.

Implementation for xBRL-XML and xBRL-CSV reports

incompleteReport: **XBRL reports MUST contain the full report, even in the case of resubmission of an amendment – no content/values from previous reports may be assumed.**

1.13 — Standalone Document Declaration

Rule

The standalone document declaration in the XML declaration (e.g.: <?xml version="1.0" encoding="UTF-8" standalone="yes" ?> or <?xml version="1.0" encoding="UTF-8" standalone="no" ?>) is only relevant for XML documents using a DTD. This information has no meaning for XBRL reports and may cause problems to some software component.

Implementation for xBRL-XML reports

standaloneDocumentDeclarationUsed: **XBRL reports SHOULD NOT use the XML standalone declaration.**

1.14 — @xsd:schemaLocation and @xsd:noNamespaceSchemaLocation

Rule

@xsd:schemaLocation and @xsd:noNamespaceSchemaLocation are attributes defined in the XML Schema specification that are used to indicate where the schema to be applied to the XML document may be found. Since the XML Schema used in xBRL-XML reports is defined by the link:schemaRef element, these attributes may introduce ambiguity.

Implementation for xBRL-XML reports

schemaLocationAttributeUsed**: @xsd:schemaLocation or**

noNamespaceSchemaLocationAttributeUsed: **@xsd:noNamespaceSchemaLocation MUST NOT be used.**

1.15 — XInclude

Rule

The XInclude specification provides a way to embed an XML document in another one, by using xi:include elements.

Implementation for xBRL-XML reports

xIncludeUsed**: XBRL reports MUST NOT use the XInclude specification (xi:include element).**

2. XBRL report syntax rules

2.1 — The existence of xml:base is not permitted

Rule

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

Implementation for xBRL-XML reports

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

2.2 — The absolute URL has to be stated for the taxonomy reference element

Rule

The taxonomy which is used for an XBRL report is identified by a URL. Although it is often convenient to work with local copies of the relevant taxonomies, it is important that these taxonomy reference elements resolve to the published entry point locations.

Note: XBRL software typically provides functionality to "remap" references to URLs of published entry points to local copies of the taxonomy.

Implementation for xBRL-XML reports

inappropriateSchemaRef: **The link:schemaRef element in submitted reports MUST resolve to the full published xsd entry point URL (absolute URL).**

Implementation for xBRL-CSV reports

InappropriateTaxonomyRef**: The extends element in submitted reports MUST resolve to the full published JSON entry point URL (absolute URL).**

2.3 — Only one taxonomy reference is allowed per report

Rule

Under the XBRL standard, a report can reference one or more taxonomies. When using the EBA taxonomy however, only a single entry point must be referred to in any report. This entry point will specify all required data points, and is used to reference a particular report type.

Implementation for xBRL-XML reports

multipleSchemaRefs: **Any xBRL-XML report MUST contain only one xbrli:xbrl/link:schemaRef element.**

Implementation for xBRL-CSV reports

multipleTaxonomyRefs: **Any xBRL-CSV report MUST contain only one documentInfo/extends element.**

2.4 —The use of link:linkbaseRef elements is not permitted

Rule

Entry points for xBRL-XML reports will be defined by means of a schema. There is no use for link:linkbaseRef elements.

Implementation for xBRL-XML reports

linkbaseRefUsed: **Reference from an report to the taxonomy MUST only be by means of the link:schemaRef element. The element link:linkbaseRef MUST NOT be used in any report document.**

2.5 —XML comments and documentation are ignored by EBA

Rule

Comments may be present in reports sent to EBA but their content will be ignored. Any information inside the report that does not get reported as a fact will be ignored by the EBA.

Implementation for xBRL-XML reports

CommentsAreIgnored: **Relevant business data MUST only be contained in contexts, units, schemaRef and facts.**

CommentsAreIgnored: **A comment MUST not have any impact on the content of a report.**

2.25 — XBRL footnotes are ignored by EBA

Rule

Footnotes may be present in reports sent to EBA but their content will be ignored.

Implementation for xBRL-XML reports

xbrlFootnotesAreIgnored: **Relevant business data MUST only be contained in contexts, units, schemaRef and facts.**

xbrlFootnotesAreIgnored: **A footnote MUST not have any impact on the regulatory content of a report.**

Implementation for xBRL-CSV reports

xbrlFootnotesAreIgnored: **Relevant business data MUST only be contained in facts, units, and documentInfo/extends.**

xbrlFootnotesAreIgnored: **A footnote MUST not have any impact on the regulatory content of a report.**

2.26 – Information about the generating software

Rule

Information about the software used to create an XBRL report may help CAs in identifying common causes for issues found in those reports.

Implementation for xBRL-XML reports

missingOrIncorrectSoftwareInformation: **Information on the software component used for production of the XBRL report SHOULD be included as an XML Processing Instruction at the beginning of the file, after the XML version and encoding declaration. It SHOULD have at least the <?instance-generator> instructions and the variables: id, version and creationdate.**

Optionally the instance-generator processing instruction may include more properties, or the XBRL report may include complementary XML comments.

Example of a valid instruction:

```
<?xml version="1.0" encoding="UTF-8"?>
<?instance-generator id="EBA Data Gen" version="2015.8.28.0" creationdate="2015-09-15T16:53:43:00+02:00"?>
```

Comments MAY also be added to provide more information. Example:

```
<!--
Generated by EBA at 2015-09-15T16:53:43+02:00
(c) 2015 EBA European Banking Authority
Data Generator Version 2015.8.28.0.
-->
```

Implementation for xBRL-CSV reports

missingOrIncorrectSoftwareInformation: Information on the software component used for production of the XBRL report SHOULD be included as a json element in the report.json file. It SHOULD use the EBA element with the name generatingSoftwareInformation and provide an id, a version and creation date. Additional information can be added through the element softwareAdditionalInfo.

Example of valid element containing information on software used:

```
{
  "documentInfo": {
        "documentType": "https://xbrl.org/CR/2021-02-03/xbrl-csv",
        "extends": [
              "http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/sbp/cir-2070-2016/2020-06-30/mod/sbpimv_con.json"
        ]
  },
  "eba:generatingSoftwareInformation": {
```

```
        "eba:softwareId": "EBA Data Generator",
        "eba:softwareVersion": "1.0",
        "eba:softwareCreationDate": "2015-09-15",
        "eba:softwareAdditionalInfo": "Additional information"
    }
}
```

Context related rules

2.6 — The length of the @id attribute should be limited to the necessary characters

Rule

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 report data. Even though there is no limitation on the length of an id attribute it is recommended to keep it as short as possible.

Implementation for xBRL-XML reports

noSemanticsinID**: Semantics SHOULD NOT be expressed in the xbrli:context/@id attribute.**

longXmlIdAttribute: **The values of each @id attribute SHOULD not be excessively long.**

2.7 — No unused or duplicated xbrli:context nodes

Rule

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

Implementation for xBRL-XML reports

(a) unusedContext**: Unused xbrli:context nodes SHOULD NOT be present in the report**.

(b) duplicateContext: **An report document SHOULD NOT contain duplicated context, unless required for technical reasons, e.g. to support XBRL streaming**.

2.8 — Identification of the subject of the report

Rule

The subject of the report must be identified.

Implementation for xBRL-XML reports

The xbrli:identifier element (value combined with the @scheme attribute allows the identification of the subject of a report[5] 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).

(a) inappropriateScheme: **XBRL-XML reports MUST use a @scheme attribute that is prescribed by the supervisor. [GFM11, p. 11]**

(b) unacceptableIdentifier: **XBRL-XML reports MUST use an identifier acceptable to the supervisor (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 CA's to the EBA, the xbrl entity identifier (scheme and value) used must**

[5] Which may or may not be conceptually identical to the submitter of a report (or the preparer of the report).

have been agreed by and registered with the EBA by the CA prior to remittance.

Implementation for xBRL-CSV reports

The identity of the subject of the report is provided through the parameter "entity" in the parameter file. This parameter must be given a qualified value, i.e. a namespace and an identity value. The namespace is the same as the scheme used in xBRL-XML. The identify value is the same as the identifier used in xBRL-XML. See section 3.6 LEI and other entity codes for more information and examples.

(a) inappropriateScheme: **xBRL-CSV reports MUST use a namespace (scheme) that is prescribed by the supervisor.**

(b) unacceptableIdentifier: **xBRL-CSV reports MUST use an identity value (identifier) acceptable to the supervisor (likely to be one recognized in their reporting system), and that corresponds to the namespace (scheme) used. [GFM11, p. 11]**

(c) **For remittance of data by CA's to the EBA, the xbrl entity identifier (scheme and value) used must have been agreed by and registered with the EBA by the CA prior to remittance.**

2.9 — Single subject per report

Rule

There can only be one conceptual subject of an XBRL report. If the content of the report deals with a group of companies, that 'group' (however defined) is the conceptual subject of the report.

Implementation for xBRL-XML reports

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

2.10 — The reference date elements reported must be valid

Rule

The elements used in XBRL reports for identifying the period they refer to (reference period) all have data type which is a union of the xs:date and xs:dateTime types. EBA will only allow periods to be identified using whole days and specified without a timezone.

Implementation for xBRL-XML reports

periodWithTimeContent: **All xbrli:period date elements MUST be valid against the xs:date data type, and**

periodWithTimezone: **reported without a timezone. [GFM11, p. 16]**

Implementation for xBRL-CSV reports

periodWithTimeContent: **The reference period parameter MUST be valid against the xs:date data type, and** periodWithTimezone: **reported without a timezone. [GFM11, p. 16]**

2.11 —The existence of xbrli:forever is not permitted

Rule

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 founder of a company 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.

Implementation for xBRL-XML reports

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

2.13 — XBRL period consistency

Rule

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 supervisory XBRL report 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 XBRL report with the XBRL period concept.

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.

Implementation for xBRL-XML reports

multiplePeriodsUsed: **All xbrl periods in a XBRL report MUST refer to the (same) reference date instant.**

nonInstantPeriodUsed: **All xbrl periods MUST be instants.**

2.14 — The existence of xbrli:segment is not permitted

Rule

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, EBA only uses the xbrli:scenario element.

Implementation for xBRL-XML reports

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

2.15 — Restrictions on the use of the xbrli:scenario element

Rule

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

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

Implementation for xBRL-XML reports

scenarioContainsNonDimensionContent: **If an xbrli:scenario 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 content. [EFM13, p. 6-8]**

Fact related rules

2.16 — Duplicate (Redundant/Inconsistent) facts

Rule

Facts are business duplicates of each other in the reporting sense if they notionally convey answers to precisely the same question. Duplicates can be complete copies (where they are truly semantically equivalent), inconsistent copies or contradictory copies.

Implementation for xBRL-XML reports

Duplicate facts are XML-XBRL syntax valid. However (whether or not their values are different) the semantic meaning may be unclear.

Item X and item Y are "duplicate facts" if and only if all the following conditions apply:
1. X is not identical to Y (not exactly the same XML node[6]), 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[7], and
4. X is P-Equal to Y[8], 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)[9], and
8. If X and Y are string items, they also have S-Equal xml:lang attributes[10].

Inconsistent facts are duplicates that are not V equal.

Duplicate facts are XML-XBRL syntax valid. However (whether or not their values are different) the semantic meaning may be unclear.

An XBRL-XML report must not have duplicated business fact items.

duplicateFactXBRL-XML: XBRL-XML reports MUST NOT contain duplicate business facts. [EFM13, p. 6-10]

Implementation for xBRL-CSV reports

---

6 This apparently trivial condition is stated here since it is sometimes relevant, e.g. when X and Y are the result of different XPath conditions

7 2&3 may loosely be considered to mean "refer to the same primary item"

8 Somewhat irrelevant in the EBA context, since all data fact items should be reported in a single root element, and no tuples are used to report data facts.

9 1-7 effectively mean "refer to the same data point". Note that this definition is very similar to, but not the same as the definition of a "duplicate item", notably it does not require that facts be U-equal to be considered "duplicate facts".

10 Multiple string facts that would otherwise be duplicates are in principle acceptable in the EBA reporting context if each has a distinct effective xml:lang attribute (i.e. if they are translations of each other). Note that the following elements do NOT make two facts non-duplicate if they differ (or if they are the same!): **value**, decimals, xml:lang for non-strings.

xBRL-CSV has a feature called “allowed duplicates”. Through this feature the taxonomy author can specify what kind of duplicates are allowed[11]. EBA has chosen to use this feature and allow complete duplicates only.

2.16.1 — No multi-unit fact sets

Rule

Two facts which differ only by unit are not technically duplicates. Indeed there might be situations in which, for example, the natural answer to a question is a bundle of set of values in several currencies (e.g. £4, $3, €3). However there is clearly a significant potential for confusion with such reporting - e.g. are the different facts supposed to be alternatives ($4 or £3), equivalents ($4 = £3), to be taken as a set ($4 and £3), or just a mistake.

In order to avoid any such doubt or confusion, reporting of “the same fact”[12] in more than one unit is not allowed in EBA reporting.

Implementation for xBRL-XML and xBRL-CSV reports

factsDifferingOnlyByUnit: **XBRL reports MUST NOT contain business facts which would be duplicates were their units not different[13].**

2.17 — The use of the @precision attribute is not permitted

Rule

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 in most uses the decimals value is likely to be one of a limited set e.g. 2, 0, -3, -6, -9 or INF (and often the same for all/many facts). Moreover, given a decimals value the precision can always be computed, but this is not symmetric.

Implementation for xBRL-XML reports

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

2.18 — Interpretation of the decimals setting

Rule

The decimals setting indicates the accuracy of the reported fact value. A numeric fact that has a decimals property with the value n is considered to be “correct to n decimal places”. Leading zeros and trailing digits should be compact and appropriate to the reported value.

[11] https://www.xbrl.org/Specification/xbrl-csv/CR-2021-02-03/xbrl-csv-CR-2021-02-03.html#511-allowed-duplicates-feature

[12] i.e. facts which meet all the conditions in rule 2.16 except point 6.

[13] OIM defines these facts as “multi-unit alternatives”. See https://www.xbrl.org/Specification/oim/CR-2021-02-16/oim-CR-2021-02-16.html#sec-multi-unit-alternatives

The EBA will interpret the decimals setting on reported data as specifying that the absolute difference between the true value of the number as known to the reporter and its reported lexical representation (known as the "absolute error" of the representation - e_{abs}) is less than or equal to 0.5×10^{-n}. Reporters must prepare submitted reports consistently with this interpretation[14].

The EBA XBRL validation rules use interval arithmetic for validation. To best enable XBRL Formula calculations to be performed on reported values for validation purposes, preferably no truncations or rounding or any other kind of change should be applied to the reported lexical representation of the numeric facts in the XBRL report. See the explanatory RFC at http://www.xbrl.org/RFC/PDU/PWD-2008-10-09/PDU-RFC-PWD-2008-10-09.html. Note however that if numbers are for any reason rounded, they MUST be rounded as per the XBRL 2.1 specification (i.e. [IEEE-754] 4.3.1 Rounding-direction attributes to nearest, roundTiesToEven), and as above the decimals setting must accurately represent the relationship between the reported and unrounded values.

Implementation for xBRL-XML reports

The decimals setting is implemented in xBRL-XML through the decimals attribute of a fact.

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

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

(c) **The reported accuracy of a numeric fact SHOULD be a realistic indication of the accuracy to which the lexical representation represents the true value. In particular it SHOULD NOT be excessively high.[15]**

Implementation for xBRL-CSV reports

The decimals setting is implemented in xBRL-CSV through the decimals property of a fact.

(a) missingDecimalsProperty: **The accuracy of a numeric fact MUST be expressed using decimals property.**

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

(c) **The reported accuracy of a numeric fact SHOULD be a realistic indication of the accuracy to which the lexical representation represents the true value. In particular it SHOULD NOT be excessively high.[15]**

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.

[14] 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

[15] E.g. decimals setting of greater than 2 would generally be inappropriate for calculated "monetary" values resulting from e.g. multiplications or divisions, "INF" is often unlikely to be appropriate for calculated values etc.

Accuracy Requirements			
Data Type	**Decimals setting**	**Note**	**Representation**
Monetary[16]	>= -3, >= -6 for the module Funding Plans only		42563.26
Percentage	>= 4	Must be expressed as a ratio in reports – 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 setting, means that in general 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 *(i.e. to nearest value) to thousands[17].*

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

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

Accuracy of the amount	Value of decimals setting
Absolutely exact monetary, percentage or other amount	INF
Accurate to millions	-6
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

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

[17] For the funding plans module the equivalent observation regarding truncating vs rounding to millions applies.

Examples: The table below illustrates correct use.

Data	Reported Value	Value of decimals setting	Range of value considered in interval arithmetic
A percentage (ratio) of (exactly) 46%	0.46	INF[18]	0.46
A profit margin of 9.3% (minimum accuracy)	0.093	4	0.09295 to 0.09305
Monetary amount “in millions”	1534512	-6	1034512 to 2034512
Monetary amount “in thousands”	117822	-3	117322 to 118322
Monetary amount “in hundreds”	124265	-2	124215 to 125215
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 report 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 report files are produced the canonical representation is used.

2.19 —Guidance on use of zeros and non-reported data

Rule

Data could be reported with a non-zero value, as zero or unreported.

Empty values are not allowed.

Implementation for xBRL-XML reports

nilUsed: **The @xsi:nil attribute MUST NOT be used for facts in the report.**

emptyUsed: **For string type metric, the empty string MUST NOT be reported.**

Implementation for xBRL-CSV reports

nilUsed: **A fact MUST not be reported as nil.**

Therefore the special value #nil MUST NOT be used for facts in the report.

emptyUsed: **A fact MUST not be empty.**

Therefore the special value #empty MUST NOT be used for facts in the report.

[18] N.B. it is only appropriate to use “INF” where the true value is known to be absolutely precisely the value reported, as written. E.g. monetary balances in cents, or chosen rather than calculated percentages.

The table below shows the different possible scenarios:

Reported Zero or Non-zero value	e.g. xBRL-XML <eba_met:mi53 unitRef="uEUR" decimals="2" contextRef="c2">1025.25</eba_met:mi53> e.g. xBRL-CSV datapoint,factValue dp31870,eba_AS:x1		The value of the fact is known.
Reported nil value	e.g. xBRL-XML <eba_met:mi53 unitRef="uEUR" contextRef="c2" @xsi:nil="true" /> e.g. xBRL-CSV datapoint,factValue dp31870,#nil		MUST NOT be used
Reported empty value	e.g. xBRL-XML <eba_met:si53 contextRef="c2"></eba_met:si53> e.g. xBRL-CSV datapoint,factValue,PBE dp439579,#empty,xyzxyz		MUST not be used
Missing fact	The fact doesn't appear in the XBRL report.	Template including this fact is reported	The value is treatable as equivalent to zero (if numeric fact) or empty (if non-numeric) by the recipient.
		No template including this fact is reported	The value is “unknown” to the recipient.

Inapplicable information need not be included in an XBRL report, 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 XBRL report (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 — Information on the use of the language setting for string facts

Rule

The language used on string based facts may need to be identified.

No restrictions are placed 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 language setting is in general not required in XBRL reports remitted to the EBA and may be omitted. It is compulsory to use the attribute in the specific case of distinguishing 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.

Implementation for xBRL-XML reports

xBRL-XML uses the xml:lang attribute to identify the language used for facts.

This attribute can be at the xbrli:xbrl element just once, or on every string based fact individually.

Implementation for xBRL-CSV reports

xBRL-CSV uses the lang property to identify the language used for facts. It has a default value of 'en' for English and can be overwritten through the parameter 'baseLanguage' in the parameter file.

Unit related rules

2.21 — Duplicates of xbrli:xbrl/xbrli:unit

Rule

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].

Implementation for xBRL-XML reports

duplicateUnit: **An XBRL report 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

Rule

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

Implementation for xBRL-XML reports

unusedUnit: **An XBRL report SHOULD NOT contain unused xbrli:unit nodes.**

2.23 — Reference unit to XBRL International Unit Type Registry (UTR)

Rule

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).

Implementation for xBRL-XML reports

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

Implementation for xBRL-CSV reports

nonUtrUnit: **The unit dimension MUST only use measures defined in the unit types registry (UTR).**

2.24 —Report of the actual physical value of monetary items (see also 3.3)

Rule

Facts that represent amounts in any currency will 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 setting).

Implementation for xBRL-XML and xBRL-CSV reports

monetaryUnitWithScaling: **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.**

3.1 – Choice of Currency for Monetary facts

Rule

In general monetary values in an XBRL report must all be expressed in the same ("reporting") currency, i.e. values should be converted to that currency.

For some specific data items however it may be indicated (in the taxonomy/DPM) that the values reported should be expressed in their "currency of denomination" (i.e. intrinsic currency), and not converted to the reporting currency[19].

This is indicated by such facts having the "Expressed in currency of denomination (not converted to reporting currency)" member of the "Currency Conversion Approach" (CCA) dimension in their context.

Such a marker will often be used in tables that e.g. report a breakdown of figures with a different currency on each sheet. Such facts should have a currency that is consistent with the currency breakdown they intend to express.

<u>One "Reporting" Currency:</u>

Implementation for xBRL-XML reports

[19] Note that this currency of denomination might of course actually be the same as the reporting currency for some facts.

(a) multipleReportingCurrencies: **An XBRL report MUST express all monetary facts[20] which do not fall under point (b) using a single currency[21]:**

"Currency of denomination" facts:

Implementation for xBRL-XML reports

(b) currencyOfDenomination: **Monetary facts whose associated context contains the eba_CA:x1 member for the CCA dimension MUST be expressed in units of their currency of denomination.**

Implementation for xBRL-CSV reports

(b) currencyOfDenomination: **Monetary facts whose associated propertyGroup contains the eba_CA:x1 member for the CCA dimension MUST be expressed in units of their currency of denomination.**

Currency dimension consistency:

Implementation for xBRL-XML and xBRL-CSV reports

(c) inconsistentCurrencyUnitAndDimension: **For facts falling under point (b), whose context also includes the dimension "Currency with significant liabilities" (CUS), the currency of the fact (i.e. unit) MUST be consistent with the value given for this dimension.**

3.2 - Non-monetary numeric units

Implementation for xBRL-XML reports

(a) pureUnitNotUsedForMonetaryValue: **An XBRL report 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 .**

Implementation for xBRL-XML and xBRL-CSV reports

(b) useDecimalFractions: **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).**

---

[20] 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). (These are likely to be encountered only in the 1.0.1 version of Benchmarking reports)

[21] For clarity – currently, where providing a breakdown by currency where the relevant data points are NOT marked as being reported in their intrinsic currency/currency of denomination, 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. Stakeholders should be aware that such tables may potentially be subject to change in future.

3.3 - Decimal representation

Implementation for xBRL-XML and xBRL-CSV reports

reportValuesAsKnownAndUnscaled: **The value of numeric facts must be expressed in the specified units, without any change of scale and should be expressed without rounding or truncation.**

The content of a numeric fact must therefore not include any scale factors like "%". Specifically, monetary values[22] 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*

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"

Wrong representation of an amount 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)

[22] Whether using monetaryItemType metrics or decimal.

3. Additional Guidance

3.4 Unused namespace prefixes

Rule

Declaring unused namespaces is uncalled for and clutters the XBRL report.

Implementation for xBRL-XML reports

unusedNamespacePrefix: **Namespace prefixes that are not used SHOULD not be declared in the XBRL report document.**

3.5 Re-use of canonical namespace prefixes

Rule

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 commonly understood prefixes used on a different namespace, or novel prefixes used for a common namespace. E.g. the prefix 'xs' used for the http://xbrl.org/2003/xbrl-instance-2033-12-31 namespace (which would normally be associated with the prefix 'xbrli', 'xs' in contrast usually being associated with http://www.w3.org/2001/XMLSchema). Note that this does not affect the use of a default namespace attribute on an element to avoid the need for the use of a namespace prefix on the element and its children altogether.

Implementation for xBRL-XML reports

notRecommendedNamespacePrefix: **Namespace prefixes, where used in XBRL reports, SHOULD mirror the namespace prefixes as defined by their schema author(s).**

3.6 LEI and other entity codes

Rule

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

Entity Code:

Before reference date 31/12/2022:

- For individual and highest-level consolidation reports this code SHOULD be the LEI of the individual/parent/reporting entity (i.e., this will be the code required and agreed by the EBA for such reports in all but very exceptional circumstances).
- For the liquidity subgroup report, this code SHOULD be the LEI of the sub group parent + ".CRDLIQSUBGRP".

From reference date 31/12/2022 onwards:

- For individual level reports, this code should be LEI + ".IND".
- For highest level consolidation report, this code should be LEI + ".CON".

- For the liquidity subgroup report, this code SHOULD be the LEI of the sub group parent + ".CRDLIQSUBGRP".
- For the other reports, this code should be agreed and preregistered with EBA.

Scheme:

- Where the LEI of the individual/parent/reporting entity is used, the scheme MUST be "http://standards.iso.org/iso/17442".
- Where the LEI + ".CON/IND/CRDLIQSUBGRP" is used (for example in a liquidity subgroup report), the scheme MUST be http://www.eba.europa.eu/eu/rs".

Implementation for xBRL-XML reports

Example for individual and highest-level consolidation report

Before the reference date 12/2022

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

From the reference date 12/2022 for the highest-level consolidation report:

```
<xbrli:entity>
  <xbrli:identifier scheme=" http://www.eba.europa.eu/eu/rs"
>LEIIDENTIFIERABCDEFG.CON</xbrli:identifier>
</xbrli:entity>
```

From the reference date 12/2022 for the individual report:

```
<xbrli:entity>
  <xbrli:identifier scheme=" http://www.eba.europa.eu/eu/rs"
>LEIIDENTIFIERABCDEFG.IND</xbrli:identifier>
</xbrli:entity>
```

where LEIIDENTIFIERABCDEFG is the appropriate LEI code for the entity.

Example for liquidity subgroup report

```
<xbrli:entity>
  <xbrli:identifier scheme="http://www.eba.europa.eu/eu/rs"
>LEIIDENTIFIERABCDEFG.CRDLIQSUBGRP</xbrli:identifier>
</xbrli:entity>
```

where LEIIDENTIFIERABCDEFG is the appropriate LEI code for the sub group parent.

Implementation for xBRL-CSV reports

The identification of the reporting entity is done using the parameter entity defined in the parameter file.

Before the reference date 31/12/2022

Example for highest-level consolidation report (section of the parameter.csv file):
name,value
entity,lei:LEIIDENTIFIERABCDEFG

where LEIIDENTIFIERABCDEFG is the appropriate LEI code for the entity and lei is the prefix for the namespace "*http://standards.iso.org/iso/17442* ".

From the reference date 31/12/2022 onwards
Example for highest-level consolidation report (section of the parameter.csv file):
name,value
entity, rs:LEIIDENTIFIERABCDEFG.CON

where LEIIDENTIFIERABCDEFG is the appropriate LEI code for the entity and rs is the prefix for the namespace " *https://eba.europa.eu/eu/rs* ".

Example for liquidity subgroup report (section of the parameter.csv file):
name,value
entity:rs:LEIIDENTIFIERABCDEFG.CRDLIQSUBGRP

where LEIIDENTIFIERABCDEFG is the appropriate LEI code for the sub group parent and rs is the prefix for the namespace "http://www.eba.europa.eu/eu/rs".

Historic errors - acceptance of variations

Please note that previous editions of these filing rules have sadly erroneously specified a scheme URI of "http://standard.iso.org/iso/17442" (note the missing final s of "standards"), in some versions solely using this form and in others the text had a mix both with and without the final s. RFC5141[23] specifies the plural form.

Given this unfortunate history of error

(a) incorrectLeiScheme: producers of XBRL reports are encouraged to switch as quickly as possible to producing the correct form "http://standard**s**.iso.org/iso/17442"

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 XBRL report will be determined by the relevant competent authority.

3.7 — Unused @id attribute on facts

Rule

Unused @id attributes on facts add no value to the supervisor and should not be included in the XBRL report unless they are valuable to the reporter.

Implementation for xBRL-XML reports

unusedFactId: **The XBRL report SHOULD NOT include unused @id attributes on facts.**

[23] https://tools.ietf.org/html/rfc5141

3.8 — Length of strings in XBRL reports

Rule

Even though there is no limitation on the length of a string reported in an XBRL report, excessively long strings are likely to cause issues in systems involved in the reporting process, many of which will have some practical constraints on the length of string they are able to handle. For this reason it is recommended to limit reported string to only the necessary characters.

Implementation for xBRL-XML and xBRL-CSV reports

excessiveStringLength: **The values of each string SHOULD be as short as possible.**

3.9 Namespace prefix declarations restricted to the document element

Rule

Namespace prefixes should be avoided in other elements of the XBRL report. This helps to improve the readability of the document and reduces its size. (See examples on page 47.)

Implementation for xBRL-XML reports

unexpectedNamespaceDeclarations: **Namespace prefixes declarations SHOULD be restricted to the document element.**

3.10 Avoid multiple prefix declarations for the same namespace

Rule

Two namespace prefixes declarations SHOULD NOT correspond to the same namespace. This helps to improve the readability of the document. (See examples on page 47.)

Implementation for xBRL-XML reports

multiplePrefixForNamespace: **Namespaces used in the document SHOULD be associated to a single namespace prefix.**

3.11 Text should not start or end with spaces

Rule

The underlying XBRL and XML specifications determine the appropriate handling of whitespace in various elements of the submitted XBRL report. In many cases, particularly string fact values, whitespace in the document is “preserved”, forming part of the value. This notably means that e.g. string typed domain values that differ only by whitespace (such as spaces or LF/CR characters, perhaps at the start or end of values) are distinct. It therefore trivially follows that such whitespace should only be included if it truly

forms part of the conveyed data (which is probably unlikely), rather than being a side effect of document layout[24].

Implementation for xBRL-XML and xBRL-CSV reports

leadingOrTrailingSpacesInText: **String facts, and string typed domain values, SHOULD not start or end with whitespace characters (i.e. MUST not do so unless, exceptionally, the whitespace is part of the data intended to be conveyed). [EIOPA15-S.2.21]**

Streaming

There is an XBRL specification called the "XBRL Streaming Extensions Module" which is under development that aims to facilitate the processing of very large XBRL reports. A "Streamable XBRL report" is an XBRL v2.1 report that obeys the serialisation constraints defined by that specification.

Several of the filing rules in this document provide guidance on the production of "nice" XBRL reports, i.e. reports that are compact, clear and less prone to errors in creation or usage. However when producing XBRL reports focussing on the efficient creation and processing of very large files it may be necessary to adapt or ignore some of these normal best practices. In general, the creation of a "streamable XBRL report" is a legitimate reason not to follow "SHOULD" rules where they conflict with or inhibit the usage of the Streaming Extensions Module specification.

Rules that are noted as being particularly relevant in this context (i.e. for which it is acknowledged that streamable XBRL reports may need not to comply) include:

- 1.6.2 —Filing indicators in several tuples
- 2.7 — No unused or duplicated xbrli:context nodes
- 2.21 — Duplicates of xbrli:xbrl/xbrli:unit

Examples

[24] Note therefore that this guidance in a sense does not actually have any significant content, it merely states that the reported values should be those intended, which is obvious. It is stated primarily to help avoid any accidental problems stemming from inclusion of additional whitespace (such as e.g. for horizontal alignment / 'pretty printing' within the xml) on the assumption that it would be irrelevant

Filing indicator usage examples

Conventions:

Positive examples are given a solid border, with crucial sections highlighted with green text and shading:

```
Sample text of example, sample text of example,
Sample text of example, crucial section of example,
Sample text of example, sample text of example
```

Key sections of counterexamples (examples of poor, discouraged or disallowed usage) are highlighted with red text and shading, and the counterexamples are given a dashed border and red background:

```
Sample text of counterexample, sample text of counterexample,
Sample text of counterexample, crucial section of counterexample,
Sample text of counterexample, sample text of counterexample
```

For xBRL-XML reports:

Consider a small module containing three templates: C_00.01, C_01.00, and C_05.01, and consider a report containing information for tables C_00.01 (mandatory template), and C_01.00 (mandatory template), but not for C_05.01. The typical approach to indicating this with filing indicator elements would be:

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="c1">C_01.00</find:filingIndicator>
  <find:filingIndicator contextRef="c1" find:filed="false">C_05.01</find:filingIndicator>
</find:fIndicators>
```

Here there is a single “fIndicators” element grouping three filing indicator elements, which indicate the intention to report the tables associated with the codes “C_00.01” and “C_01.00”, and the intention not to report the tables associated with the code “C_05.01”

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

```
<find:fIndicators>
  <find:filingIndicator contextRef="c2">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="c2" find:filed="true">C_01.00</find:filingIndicator>
  <find:filingIndicator contextRef="c1" find:filed="false">C_05.01</find:filingIndicator>
</find:fIndicators>
```

Or utilising more than one containing “fIndicators” element:

```
<find:fIndicators>
  <find:filingIndicator contextRef="A" find:filed="true">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="A">C_01.00</find:filingIndicator>
</find:fIndicators>
…
<find:fIndicators>
  <find:filingIndicator contextRef="c1" find:filed="false">C_05.01</find:filingIndicator>
</find:fIndicators>
```

Unacceptable variations include, for example:

Not indicating that a reported template is reported (C_01.00 is missing):

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="c1" find:filed="false">C_05.01</find:filingIndicator>
</find:fIndicators>
```

Not indicating that a template is not being reported (C_05.00 is missing):

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="c1">C_01.000</find:filingIndicator>
</find:fIndicators>
```

Indicating that an unreported template is reported (C_05.01 is not reported):

```
<find:fIndicators>
  <find:filingIndicator contextRef="c1">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="c1">C_01.00</find:filingIndicator>
  <find:filingIndicator contextRef="c1">C_05.01</find:filingIndicator>
</find:fIndicators>
```

Duplicating a filing indicator. Here both C_00.01 and C_01.00 appear twice, either repetition is an error, i.e. it does not matter that the two C_01.00 filing indicators are in different tuples:

```
<find:fIndicators>
  <find:filingIndicator contextRef="A" find:filed="true">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="A">C_00.01</find:filingIndicator>
  <find:filingIndicator contextRef="A">C_01.00</find:filingIndicator>
</find:fIndicators>
…
<find:fIndicators>
  <find:filingIndicator contextRef="A">C_01.00</find:filingIndicator>
  <find:filingIndicator contextRef="c1" find:filed="false">C_05.01</find:filingIndicator>
</find:fIndicators>
```

For xBRL-CSV reports:

The filing indicator declaration must be conveyed via file FilingIndicators.csv, to

if we indicate that templates C 00.01 and C 01.00 are reported, and C 05.01 is not reported

```
templateID,reported
C_00.01, true
C_01.00, true
C 05.01, false
```

Unacceptable variations include for example:

Not indicating that a reported template is reported (C_01.00 is missing):

```
templateID, reported
C_00.01, true
C_05.01, false
```

Not indicating that a template is not being reported (C_05.00 is missing):

```
templateID, reported
C_00.01, true
C_01.00, true
```

Namespace prefix declaration examples

For xBRL-CSV reports:

The namespace prefix declaration is included in the JSON meta data included in the taxonomy, no namepace prefix should be declared in anything file in the reporting package.

For xBRL-XML reports:

As shown in the example below, namespace prefix declarations should only be in the document element.

```
<?xml version="1.0" encoding="UTF-8"?>
<xbrli:xbrl xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
    xmlns:link="http://www.xbrl.org/2003/linkbase"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xbrli="http://www.xbrl.org/2003/instance"
    xmlns:eba_dim="http://www.eba.europa.eu/xbrl/crr/dict/dim"
    xmlns:eba_BA="http://www.eba.europa.eu/xbrl/crr/dict/dom/BA"
    xmlns:eba_MC="http://www.eba.europa.eu/xbrl/crr/dict/dom/MC"
    xmlns:eba_OF=”http://www.eba.europa.eu/xbrl/crr/dict/dom/OF”
    … >
    <link:schemaRef xlink:type="simple"
xlink:href="http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/its-2013-02/2013-12-
01/mod/corep_con.xsd"/>
    <xbrli:context id="i10416092">
        <xbrli:period>
            <xbrli:instant>2014-03-31</xbrli:instant>
        </xbrli:period>
        <xbrli:scenario>
            <xbrldi:explicitMember dimension="eba_dim:BAS">eba_BA:x9</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:EXC">eba_EC:x15</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:MCY">eba_MC:x195</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:PRP">eba_PL:x11</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:TCP">eba_CP:x27</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:TRI">eba_TR:x4</xbrldi:explicitMember>
                …
    </xbrli:context>
```

No namespaces should be declared on another level than the document level. The following example shows bad practice with the declaration of eba_dim at context level.

<?xml version="1.0" encoding="UTF-8"?>

```
<xbrli:xbrl xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
    xmlns:xbrli="http://www.xbrl.org/2003/instance"
    xmlns:link="http://www.xbrl.org/2003/linkbase"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:eba_BA="http://www.eba.europa.eu/xbrl/crr/dict/dom/BA"
    xmlns:eba_MC="http://www.eba.europa.eu/xbrl/crr/dict/dom/MC"
    xmlns:eba_OF=”http://www.eba.europa.eu/xbrl/crr/dict/dom/OF”
    … >
    <link:schemaRef xlink:type="simple"
        xlink:href="http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/its-2013-02/2013-12-
01/mod/corep_con.xsd"/>
    <xbrli:context xmlns:eba_dim="http://www.eba.europa.eu/xbrl/crr/dict/dim" id="i10416092">
        <xbrli:period>
            <xbrli:instant>2014-03-31</xbrli:instant>
        </xbrli:period>
        <xbrli:scenario>
            <xbrldi:explicitMember dimension="eba_dim:BAS">eba_BA:x9</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:EXC">eba_EC:x15</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:MCY">eba_MC:x195</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:PRP">eba_PL:x11</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:TCP">eba_CP:x27</xbrldi:explicitMember>
            <xbrldi:explicitMember dimension="eba_dim:TRI">eba_TR:x4</xbrldi:explicitMember>
                …
    </xbrli:context>
```

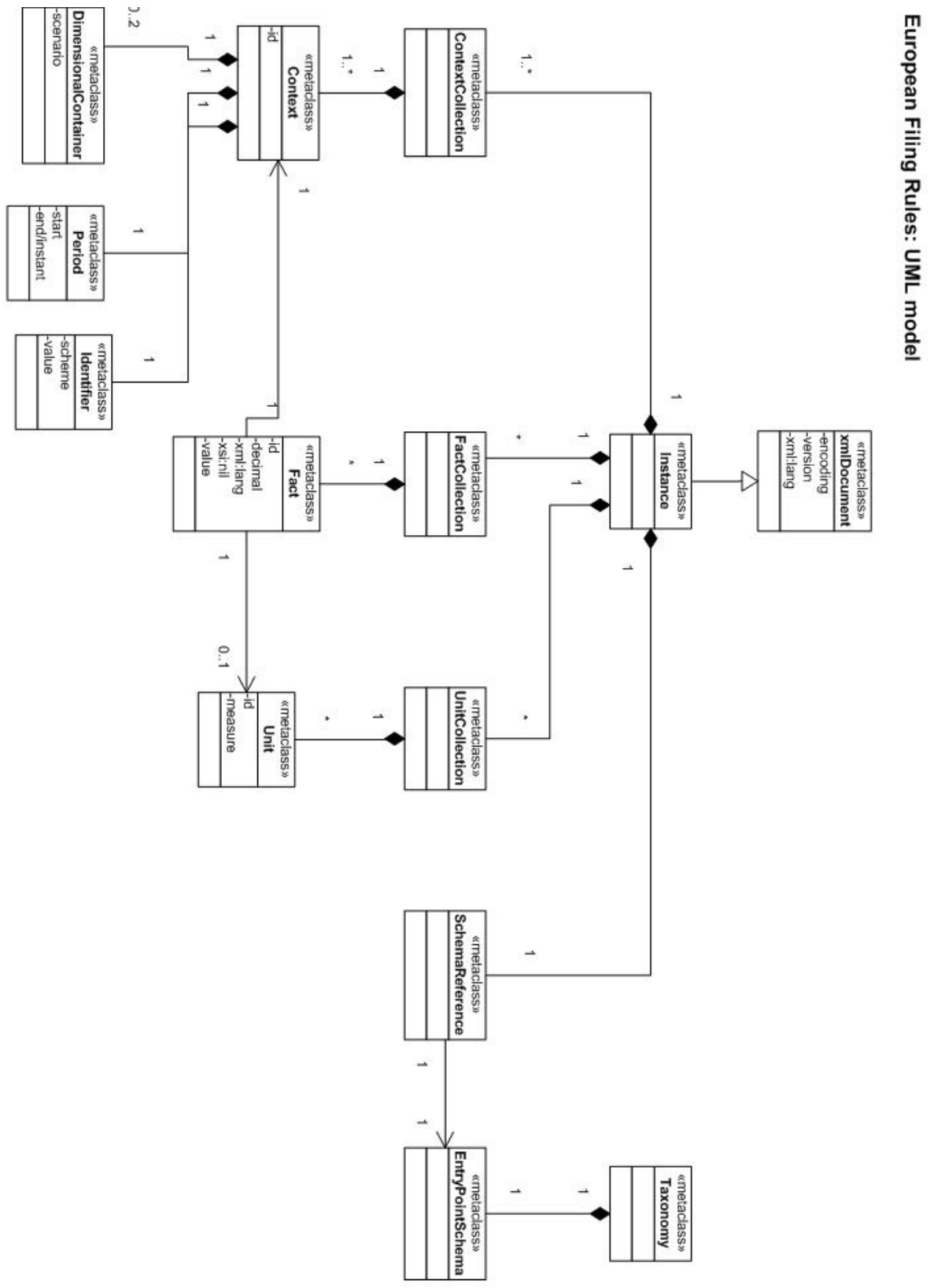
In this second wrong example the default prefix is redefined in the schemaRef element.

```
<?xml version="1.0" encoding="UTF-8"?>
<xbrl xmlns ="http://www.xbrl.org/2003/instance"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:eba_dim="http://www.eba.europa.eu/xbrl/crr/dict/dim"
    xmlns:eba_BA="http://www.eba.europa.eu/xbrl/crr/dict/dom/BA"
    xmlns:eba_MC="http://www.eba.europa.eu/xbrl/crr/dict/dom/MC"
    xmlns:eba_OF="http://www.eba.europa.eu/xbrl/crr/dict/dom/OF"
    … >
    <schemaRef xmlns="http://www.xbrl.org/2003/linkbase"
        xlink:type="simple" xlink:href="http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/its-
2013-02/2013-12-01/mod/corep_con.xsd"/>
    <context id="i10416092">
        <period>
            <instant>2014-03-31</instant>
        </period>
        <scenario>
            <explicitMember dimension="eba_dim:BAS">eba_BA:x9</explicitMember>
            <explicitMember dimension="eba_dim:EXC">eba_EC:x15</explicitMember>
            <explicitMember dimension="eba_dim:MCY">eba_MC:x195</explicitMember>
            <explicitMember dimension="eba_dim:PRP">eba_PL:x11</explicitMember>
            <explicitMember dimension="eba_dim:TCP">eba_CP:x27</explicitMember>
            <explicitMember dimension="eba_dim:TRI">eba_TR:x4</explicitMember>
                …
    </context>
```

There should be no multiple prefix declarations for the same namespace.
In the wrong example below the xbrl instance namespace is declared by the default prefix and the xbrli prefix.

```
    <?xml version="1.0" encoding="UTF-8"?>
    <xbrli:xbrl xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:link="http://www.xbrl.org/2003/linkbase"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns="http://www.xbrl.org/2003/instance"
        xmlns:eba_dim="http://www.eba.europa.eu/xbrl/crr/dict/dim"
        xmlns:eba_BA="http://www.eba.europa.eu/xbrl/crr/dict/dom/BA"
        xmlns:eba_MC="http://www.eba.europa.eu/xbrl/crr/dict/dom/MC"
        xmlns:eba_OF="http://www.eba.europa.eu/xbrl/crr/dict/dom/OF"
        … >
```


European Filing Rules: UML model
«metaclass» xmlDocument
-encoding
-version
-xml:lang
«metaclass» Instance
«metaclass» ContextCollection
«metaclass» Context
-id
«metaclass» DimensionalContainer
-scenario
«metaclass» Period
-start
-end/instant
«metaclass» Identifier
-scheme
-value
«metaclass» FactCollection
«metaclass» Fact
-id
-decimal
-xml:lang
-xsi:nil
-value
«metaclass» UnitCollection
«metaclass» Unit
-id
-measure
«metaclass» SchemaReference
«metaclass» EntryPointSchema
«metaclass» Taxonomy

File naming structure for remittance to the EBA

General

The XBRL report submitted to the EBA should be zipped, the naming structure for this zip file is as follows:
ReportSubject_Country_TaxonomyVersion_Module_ReferenceDate_CreationTimestamp.zip

zip file must contain only one XBRL file, whose naming structure for remittance to the EBA is as follows:
ReportSubject_Country_TaxonomyVersion_Module_ReferenceDate_CreationTimestamp.xbrl

ReportSubject	Before the reference date 31/12/2022: For individual and consolidated reports, it refers to Legal Entity Identifiers. For example, 549300I84DXMIK4UUL30 for Catalunya Banc For Liquidity subgroup reports, it refers to LEI + ".CRDLIQSUBGRP". For example 549300I84DXMIK4UUL30.CRDLIQSUBGRP From the reference date 31/12/2022 onwards: • For individual reports, it refers to LEI + ".IND" • For highest consolidated reports, it refers to LEI + ".CON" • For Liquidity subgroup reports, it refers to LEI + ".CRDLIQSUBGRP" • For the other types of report, the code should be agreed with EBA and preregistered with EBA
Country	ISO Country Code. For example DE for Germany
TaxonomyVersion	Framework name defined by the DPM/XBRL taxonomy in uppercase followed by the taxonomy version in 6 digits FRAMEWORKNAMEXXYYZZ. For example for the COREP reporting taxonomy 2.0.1: XX=02,YY=00 and ZZ=01 → COREP020001
Module	Before the reference date 31/12/2022 Module name as defined by the taxonomy without underscore and in upper-case. For example for the module corep_lcr_con defined by the taxonomy →COREPLCRCON For Liquidity subgroup reports, it refers to the relevant consolidated module name in upper-case. For example COREPLCRDACON From the reference date 31/12/2022 onwards, Module name as defined by the taxonomy without underscore and in upper-case. For example for the module corep_lcr defined by the taxonomy →COREPLCR (Modules defined by the taxonomy do not contain any consolidation/individual information anymore).
Reference Date	YYYY-MM-DD. For example: 2012-03-31
Creation Timestamp	YYYYMMDDhhmmssfff. For example, 20140602581112463

xBRL-XML reports

The zip file must contain only one XBRL file, whose naming structure for remittance to the EBA is as follows: ReportSubject_Country_TaxonomyVersion_Module_ReferenceDate_CreationTimestamp.xbrl

A typical XBRL report file created by a CA and conforming to the above file naming structure will be named as follows:

Before the reference date 31/12/2022:

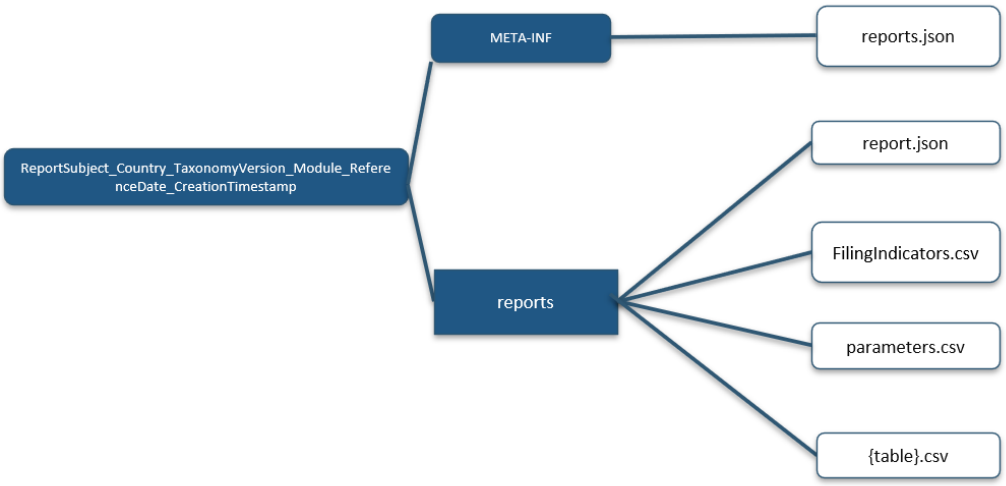
635400PNXCHKON18BK07_AT_COREP030000_COREPOFCON_2021-06-30_20210904181132453.xbrl

From the reference date 31/12/2022 onwards:

635400PNXCHKON18BK07.CON_AT_COREP030100_COREPOF_2022-12-31_20230604181132453.xbrl

xBRL-CSV reports

The EBA xBRL-CSV reporting package respects the Report Package 1.0 (xbrl.org)[25], the content of this zip package consist of:



The root folder must be named as the zip file: ReportSubject_Country_TaxonomyVersion_Module_ReferenceDate_CreationTimestamp.

It contains two folders:

1. META-INF: it contains one file reports.json with a fixed content:

[25] https://www.xbrl.org/Specification/report-package/PWD-2020-12-09/report-package-PWD-2020-12-09.html

```
{
  "documentInfo": {
   "documentType": "http://xbrl.org/PWD/2020-12-09/report-package"
  }
}
```

2. reports folder: it contains a set of files:

   a. report.json file

```
{
    "documentInfo": {
        "documentType": "https://xbrl.org/CR/2021-02-03/xbrl-csv",
        "extends": [
            "http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/if/its-002-2021/2021-05-08/mod/if_class2_con.json"
        ]
    }
}
```

   "The "extends" key must point to a list with only one value and this value must resolve to the published, full **json entry point** URL (absolute URL).

   b. parameters.csv file

```
name,value
entityID,lei:DUMMYLEI123456789012
refPeriod,2021-09-30
baseCurrency,iso4217:EUR
decimalsInteger,0
decimalsMonetary,-3
decimalsPercentage,4
decimalsDecimal,2
```

   The first line of this csv file is fixed as: name, value. Reporters must provide values for entityID, refPeriod and baseCurrency. Reporters must fill the decimals parameter for the types of metrics used in the reported module.

   c. FilingIndicators.csv file

```
templateID,reported
C_18.00,true
C_19.00,false
C_20.00,true
C_21.00,false
C_22.00,true
```

   This file is used to indicate which templates are reported.

   d. {table}.json (ex: i_08.02.csv)

```
datapoint,factValue,CPJ,ICO
dp459894,hulr qokp,yvi mamkt,eba_BT:x15
dp459967,eba_ZZ:x1,yvi mamkt,eba_BT:x15
dp459994,520906.83,yvi mamkt,eba_BT:x15
dp460107,102851.02,yvi mamkt,eba_BT:x15
dp459894,hpfz ywam,fpl cshpu,eba_BT:x16
dp459967,eba_ZZ:x1,fpl cshpu,eba_BT:x16
dp459994,262382.42,fpl cshpu,eba_BT:x16
dp460107,568874.59,fpl cshpu,eba_BT:x16
```
[26]

The first line of this csv file is fixed as: datapoint, factValue and open dimension name or typed dimension name if they exist for this table. For example in the table i_08.02.csv, there are one typed dimension CPJ and one open dimension ICO.

Multi-currency XBRL reports

Use of member eba_CU:x46

The member eba_CU:x46 ("Other Currency (open axis tables)") is intended to provide flexibility for rare edge cases.[27]

For facts that are indicated as being ideally reported denominated in their underlying currency (rather than converted to a common reporting currency for the report) eba_CU:x46 - is essentially *consistent* with (or more precisely not inconsistent with) the usage of any currency unit for the reported fact. Situations in which it would be appropriate to use this member would include, for example:

- If it should happen that the EBA enumerated currency list (the CU domain) is out of sync with the ISO list, then it could be used with values in an iso4217 currency which is not listed in the EBA CU dimension.

- If there is for any period a real world currency which is not yet iso4217 recognized, or in the case of currently existing currencies or cryptocurrencies for which there is no iso4217 code, then it could, if required, be used to report these values. The actual reported figures will need to be expressed as their value in some iso4217 currency of course, ideally the main reporting currency for the report, since the XBRL specification requires monetary facts to use iso4217currency units.

More complex situations (such as multiple such unavailable currencies being reported) may generally be handled by reporting equivalent combined values (preferably in the reporting currency) under this member.

[26] According to xBRL-CSV: mapping from Open Information Model 1.0, if a string contains a comma, carriage return, linefeed or double quote symbol, then the sting value must be enclosed in double quotes.

[27] Usage of this entry would of course be likely to require the conveyance of an explanation of the situation in parallel to the reported instance itself.

Checking of appropriate currency usage (implementing rule 3.1)

To elaborate on rule 3.1 consider the following outline of a possible approach to checking and enforcing this:

1) Determine a "reporting currency" for the XBRL report. This should be the currency of (the unit of) any reported fact which uses a metric with the data type "Monetary" and does not have eba_CA:x1 as a value for the CCA dimension in its context.
2) Check that all other monetary facts without eba_CA:x1 as a value for the CCA dimension in their context use (units with) this same currency. If not, there is a breach of filing rule EBA 3.1 (a) – Only one primary reporting currency may be used (multipleReportingCurrencies).
3) For all facts with eba_CA:x1 as a value for the CCA dimension, and which have a value for the CUS dimension
    a. If the value is an eba_CU member with a three alpha character code (e.g. USD, GBP, ALL etc.) then ensure the currency of the fact matches this value. If not there is a breach of filing rule EBA 3.1 (c) – The unit currency of facts expressed in currency of denomination must be consistent with the value given for their currency dimension (inconsistentCurrencyUnitAndDimension).
    b. If the value is eba_CU:x0, ensure that the currency of the fact matches the "reporting currency" from point 1 (where determined, or at least all the other facts in this clause) . If not, there is a breach of filing rule EBA 3.1 (c) – The unit currency of facts expressed in currency of denomination must be consistent with the value given for their currency dimension (inconsistentCurrencyUnitAndDimension). This is because any "total/all currency" figures must be expressed in the primary reporting currency.
    c. If the value is eba_CU:x46 ("Other Currency (open axis tables)"), accept[28] any XBRL acceptable currency for the fact. Note/warn about the usage (nonSpecificCurrencyDimensionUsed).
4) For all facts with eba_CA:x1 as a value for the CCA dimension, and which do NOT have a value for the CUS dimension, accept any XBRL acceptable currency for the fact.

[28] Subject of course to any other relevant technical, semantic or regulatory constraint, for example the need to continue to ensure rule 2.16.1 — No multi-unit fact sets (factsDifferingOnlyByUnit) is respected.