Representation of the Data Point Model (adjusted to support DPM Refit) in the format of XBRL Taxonomy

This document describes the design and approach applied by the European Banking Authority [EBA], the European Insurance and Occupational Pensions Authority [EIOPA], the European Central Bank [ECB], the Single Resolution Board [SRB] and some National Competent Authorities [NCAs] to represent DPM models (following changes resulting from the DPM Refit project) using semantic and syntax of XBRL taxonomies. It also describes modularisation of the XBRL taxonomy content in folders and files, applied naming conventions, etc.

Working Draft

Revised 04/2023

Table of Contents

1 Introduction

This document presents and explains the architecture of the XBRL taxonomy applied by the EBA, EIOPA and other European or National Competent Authorities.

The expected direct audience of this document are software developers working on regulatory reporting solutions utilizing the EBA or EIOPA XBRL Taxonomies by the national competent authorities [NCAs] required to pass supervisory data to the EBA or EIOPA. Additionally, given the possibility of this taxonomy forming, to some degree, the basis for reporting to some national competent authorities, it will also be of software vendors or developers involved in the regulatory reporting process in the European Union or other scopes.

1.1 Relation to previous works

This document includes modifications to the currently applied by the EBA and EIOPA architecture of the XBRL taxonomies that aimed at facilitating data exchange of the upcoming DPM Refit evolution. These changes are necessary to address modifications in modelling introduced by the DPM Refit, to cope with some critic inefficiencies or missing functionalities in the current DPM, the lack of a mechanism for historization of certain concepts.

In addition, the architecture was stripped down of some artefacts that seemed unnecessary, such as normative codes for framework taxonomies. There are also a few other improvements to simplify the XBRL representation of the model as well as to unify architectures of EBA and EIOPA taxonomies which until now had a few specific flavours requiring vendors to do customizations in their tools.

Importantly, XBRL taxonomies of EBA and EIOPA created under this updated architecture remain compliant with normative XBRL specifications and use functionalities as provided by the standard.

It is expected that this architecture is applied in the EBA and EIOPA XBRL taxonomies as soon as it is feasible and independently from the DPM Refit as current DPM models can be easily represented in that format.

It is expected that both EBA and EIOPA taxonomies in the new architecture are produced automatically from the DPM Refit models. The metadata management and modelling solution for the authorities is expected to be *almost* fully harmonised.

2   Assumptions

DPM (including DPM Refit) metamodel enables comprehensive modelling of metadata for the purposes of its management and further use in various scenarios, one of which is the support of data exchange. As a result, XBRL taxonomies reflecting DPM models shall include only this information that is strictly necessary to enable exchange of date. In other words, there can be information in the DPM models that is not available in taxonomies. It can be accessed if needed (e.g. by vendors to enhance their solutions), for example by linking to business codes of concepts that are mapped from the model to the XBRL representation.

3   Relation to standards and other documents

Comprehension of the Extensible Business Reporting Language (XBRL) 2.1 Specification[1] and various other XBRL Specifications such as XBRL Dimensions 1.0, XBRL Formula 1.0, Generic Link 1.0, Table Linkbase 1.0, Extensible Enumerations 1.0/2.0, OIM 1.0 is required to understand the content of this document.

For modelling of data (in terms of methodology and format) as well as physical representation in XBRL syntax, the EBA and EIOPA followed the approaches applied for various deliverables of the Eurofiling project[2].

In particular, the EBA and EIOPA applied the Data Point Modelling methodology and the Data Point Model [DPM] format to the description of the exchanged data[3].

The mapping of this DPM to an XBRL taxonomy follows the general architectural approach of the preliminary FINREP taxonomies published on the Eurofiling website[4], EBA and EIOPA websites[5], an approach shared also with the similar solutions developed by various NCAs.

4   XBRL specifications compliance

Following the XBRL standard requirements, the EBA and EIOPA taxonomies, and any XBRL instance documents are compliant with the XBRL 2.1 specification as of December 31, 2003 with Errata Corrections up to February 20, 2013, and the Dimensions 1.0 specification as of September 18, 2006 with errata corrections up to January 25, 2012.

The business rules layer in the form of linkbase files is defined according to the XBRL Formula Specification 1.0 - 2009 – 2016 and supporting specifications (Registry – 2009-2011, Generic Links – June 22, 2009). Assertion test expressions or filters may also use XPath/XQuery and XBRL Functions.

---

[1] https://specifications.xbrl.org/specifications.html

[2] Eurofiling is an open joint initiative collaborating with the EBA, ECB, EIOPA, ESMA, SRB and many other stakeholders in the regulatory space. All deliverables of the Eurofiling project can be found on https://www.eurofiling.info

[3] https://2022.eurofiling.info/past-events/dpm-refit/

[4] https://www.eurofiling.info/finrepTaxonomy/EBA-DPM-XBRL-Mapping.pdf

[5] https://www.eiopa.europa.eu/tools-and-data/supervisory-reporting-dpm-and-xbrl_en

Rendering of tables is created according to Table Linkbase specification published on March 18, 2014 with Errata corrections up to July 17, 2018.

For enumerated metrics' dropdowns, the taxonomy utilizes the Extensible Enumerations 1.0 specification from October 29, 2014 and/or Extensible Enumerations 2.0 from February 12, 2020.

For clarity of this document, XBRL technical constructs referenced in various sections are identified by their qualified names [QNames]. Prefixes applied in these QNames to abbreviate the namespaces follow the canonical namespace prefixes as presented in *Table 1*.

Table 1. Prefixes and namespaces of the XBRL technical files referenced in this document.

Prefix	Namespace
df	http://xbrl.org/2008/filter/dimension
enum	http://xbrl.org/2014/extensible-enumerations
gen	http://xbrl.org/2008/generic
iso4217	http://www.xbrl.org/2003/iso4217
label	http://xbrl.org/2008/label
link	http://www.xbrl.org/2003/linkbase
nonnum	http://www.xbrl.org/dtr/type/non-numeric
num	http://www.xbrl.org/dtr/type/numeric
sev	http://xbrl.org/2016/assertion-severity
table	http://xbrl.org/2014/table
tp	http://xbrl.org/2016/taxonomy-package
variable	http://xbrl.org/2008/variable
xbrldi	http://xbrl.org/2006/xbrldi
xbrldt	http://xbrl.org/2005/xbrldt
xbrli	http://www.xbrl.org/2003/instance
xlink	http://www.w3.org/1999/xlink
xs	http://www.w3.org/2001/XMLSchema
xfi	http://www.xbrl.org/2008/function/instance
xff	http://www.xbrl.org/2010/function/formula
fi	http://www.xbrl.org/taxonomy/int/filing-indicators/REC/2021-02-03

5   Publication and distribution

For convenience, the EBA and EIOPA taxonomies are distributed as a package according to the Taxonomy Packages 1.0 specification (as of April 19, 2016). This allows users to quickly identify relevant entry points and enables software to automatically configure the necessary remappings.

6 Supporting concepts

This chapter describes some concepts to facilitate the definition of the mapping rules between the abstract Data Point Model and XBRL taxonomies.

6.1   Model supporting schema and other technical files

The XBRL representation of the model makes use of some schema definitions in the namespace *http://www.eurofiling.info/xbrl/ext/model*. The official location of this schema file is *https://www.eurofiling.info/eu/fr/xbrl/ext/model.xsd[6]*. Throughout this document, the prefix *model* will be used to refer to this schema namespace (see *Table 2*).

The *model.xsd* schema contains definitions of dimensional constructs and linkbase placeholders to increase validation of reports, for instance for superfluous, unwanted content (in particular to prevent default use of metrics (i.e., when not explicitly allowed) and block scenario and segment for filing indicators). It also contains various constructs that provide additional information on the XBRL items defined in the taxonomy and their relationships specific to DPM approach. For instance, the attribute *fromDate* describes the reference date the item is valid from, the *arcrole http://www.eurofiling.info/xbrl/arcrole/applies-to-table* describes to which table the assertions in an assertion set are related to and the *roleType http://www.eurofiling.info/xbrl/role/rc-code* is used to identify a label as a row-column-code. DPM Refit introduces versions of enumerations, therefore new *arcroles* are added to describe how versions are related.

Apart from the *model.xsd* schema, *http://www.eurofiling.info/eu/fr/xbrl/ext* folder includes also other technical files explained in the next sections of this document. One of these files is *filing-indicators.xsd* schema associated with *filing-indicators-def.xml, filing-indicators-check.xml,   filing-indicators-check-err-en.xml   and   filing-indicators-check-lab-en.xml* where filing indicators are assigned with an empty hypercube to block the use of *xbrli:segment* and *xbrli:scenario* in the context they refer to and assertions ensuring that filing indicators are declared in the report and they are used in the required tuple or typed dimension structure.

Another construct defined in referenced *http://www.eurofiling.info/xbrl/ext* folder is a pivot variable declared in *pivot-variable.xml* that supports the definition of existence checks using value assertions and a set of XBRL custom functions' definitions (for example   *interval-arithmetics.xml,   isin-check.xml,   math.xml*)   referenced   by *https://www.eurofiling.info/eu/fr/xbrl/func/func.xsd.* XBRL Formula assertions may also

---

[6] It can also be accessed through the XBRL Taxonomies Registry: https://taxonomies.xbrl.org/

use constructs defined in linkbases placed in the *http://www.eurofiling.info/eu/fr/xbrl/val* folder.

Table 2. Prefixes and namespaces of the model supporting schema and other technical files used in this document.

Prefix	Namespace
model	http://www.eurofiling.info/xbrl/ext/model
find	http://www.eurofiling.info/xbrl/ext/filing-indicators
iaf	http://www.eurofiling.info/xbrl/functions/interval-arithmetics
isin_fn	http://www.eurofiling.info/xbrl/functions/isin[7]
lei-fn	http://www.xbrl.org/taxonomy/int/lei/2020-07-02/functions[8]
math_fn	http://www.eurofiling.info/xbrl/functions/math[9]
pvar	http://www.eurofiling.info/xbrl/ext/pivot-variable

Schema and linkbase files described in this section are imported or referred from various XBRL taxonomy files.

6.2 Public elements

Public elements are all concepts of the model that are identified by a code in a certain scope and may include some additional information such as readable labels, definitions and legal references in different languages.

6.2.1 Standard labels

Language specific information of public elements is represented using the following label resources:

– XBRL 2.1 labels (*link:label*) for *xbrli:item*s (or derived) public elements,

– generic labels (*label:label*) for public elements represented as XLink resources or other constructs (e.g. *link:roleType*s).

In general, the default (standard) role (*http://www.xbrl.org/2003/role/link*) is used for extended links containing the label resources; however, some specific labels may be assigned also in different extended link roles (e.g. domain member labels specific to hierarchies as explained in section "Explicit domain members and their relationship")

The role types used as roles for generic and standard label resources are lister in *Table 3*.

---

[7] Available by importing http://www.eurofiling.info/eu/fr/xbrl/func/func.xsd
[8] See https://www.xbrl.org/guidance/lei-taxonomy-guidance/ for guidance on checking LEI's
[9] Available by importing http://www.eurofiling.info/eu/fr/xbrl/func/func.xsd

Table 3. Role types used as roles for generic and standard label resources.

Property	Generic label role	Standard label role
Standard name	http://www.xbrl.org/2008/role/label	http://www.xbrl.org/2003/role/label
Definition	http://www.xbrl.org/2008/role/-verboseLabel	http://www.xbrl.org/2003/role/verboseLabel
Legal references[10]	http://www.xbrl.org/2008/role/-documentation	http://www.xbrl.org/2003/role/documentation

The labels for the concepts of a schema or a linkbase file are placed in a separate label linkbase file for each distinct language, located in the same folder as its corresponding schema or linkbase file.

The naming convention for these label linkbase files is: *{base file name}-lab-{lang}.xml* where *{base file name}* is the name of the schema or linkbase file where the concept is defined (without extension) and the *{lang}* component is the ISO 639-1 code of the language (lowercase).

In case of needing any region or country code to identify more specifically the language, the following notation shall be used:

> {base file name}-lab-{lang}-{country}.xml

Where {country} corresponds to the ISO 639-2 code of the region or country (lowercase).

The primary language for the EBA and EIOPA XBRL taxonomies is English (ISO 639-1 code "en").

6.2.2 Specific labels

In addition, some concepts may require a special linkbase to represent specific labels needed for different purposes (e.g. codes to be used as filing indicators' values). The names of these linkbase files are constructed as follows: *{base file name}-lab-{lang}-codes.xml* or *{base file name}-lab-codes.xml*

The labels for these codes are represented as resources with a custom role. In particular, the role defined in the Eurofiling *model.xsd* schema for resources representing codes for filing indicators is *http://www.eurofiling.info/xbrl/role/filing-indicator-code* while the role for resources representing the table row/column/sheet codes is *http://www.eurofiling.info/xbrl/role/rc-code*.

Hierarchy nodes specific labels are defined in the hierarchy extended link role in a separate file for each domain.

---

[10]   Current references are described in plain English; as a consequence, labels are a better solution than reference linkbases. In the future, a structured approach for legal references could be undertaken.

Extensions might use the same mechanism to add their own application specific codifications using different roles.

7 Logical taxonomy architecture

This section describes in detail the components and content of the taxonomy. The diagram provided in *Annex 3. EBA and EIOPA XBRL Taxonomy: Owners, Folders, Files, Namespaces and Prefixes.* may be helpful for the comprehension of this section.

7.1 Owners

The owner represents a location and a namespace in which a set of related concepts are defined. The owner is closely related to the idea of extensibility in XBRL. The main properties of the owner are:

– namespace (*{ons}*),

– prefix (*{opre}*), and

– official location (*{oloc}*).

The owner's namespace is a URI used to define the namespace used by the concepts. The prefixes associated to the namespaces in the taxonomy's files and the associated documentation are called "canonical prefixes". Items of the DPM and the taxonomy are referenced by their QName, using their canonical prefix.

Official location is a URL used to specify the location where taxonomy files associated with that owner are to be published. Different owners must have different official locations, even if owners share a single internet domain. The official location of the taxonomy should be built from the internet domain of the institution plus a component representing the geographical area covered by the institution (as *eu* for EIOPA artefacts) followed by the identification of the type of standard used to express information requirements (e.g. *xbrl*).

Examples of owner namespaces and locations are presented in *Table 4*.

Table 4. Examples of owner namespaces and locations.

Owner	Namespace	Official location	Prefix
Eurofiling	http://www.eurofiling.info/xbrl	http://www.eurofiling.info/eu/fr/xbrl	eu
EIOPA	http://eiopa.europa.eu/xbrl	http://eiopa.europa.eu/eu/xbrl/	eiopa
EBA	http://www.eba.europa.eu/xbrl/crr	http://www.eba.europa.eu/eu/fr/xbrl/crr	eba

Note: according to the underlying DPM, the EIOPA model is defined in two versions[11]: highly dimensional (HD) and moderately dimensional (MD). In general, the difference between the two is the definition of metrics that in the HD version represents very basic data types while in the MD version metrics include additionally some dimensional information while other dimensional properties are shared (reused in both versions). On

---

See section IV.2 of EIOPA DPM Documentation_2.8.0.pdf

the technical level EIOPA information requirements are defined in the XBRL taxonomy only in the MD approach.

8 Dictionary layer

Dictionary layer contains the definition of business properties identified in the DPM Dictionary. The properties can subsequently be used in identification of currently requested information requirements.

8.1 Core concepts

The core concepts of the dictionary are metrics, dimensions, domains and domain members.

All the concepts in the dictionary are public elements.

The core concepts are never deleted[12]. As a result, the dictionary will grow in time as the new concepts are added.

All files except enumerated metrics' files in the dictionary of concepts are placed under the folder "*dict*" in the official location {oloc} of its owner. Its namespace is obtained by adding a suffix that depends on the type of element to the namespace of the owner {ons}. The prefix to represent that namespace is obtained by adding a predefined suffix to the prefix of its owner *{opre}* where *{oloc}*, *{ons}* and *{opre}* are defined as in 7.1 7.1Owners, and *{dc}/{DC}* is the code of a domain in lower and upper case respectively.

For enumerated metrics' files, they are placed under folder dict/met/{version-number}/ in the official location of its owner, its namespace prefix is {opre}_met_{version-number}[13] (ex: eba_met_3.4.0).

Dictionary concept	Official location	Target namespace	Namespace prefix
Metrics (non-enumerated)	{oloc}/dict/met/met.xsd	{ons}/dict/met	{opre}_met
Metrics (enumerated)	{oloc}/dict/met/{version-number}/met.xsd	{ons}/dict/met/{version-number}	{opre}_met_{version-number}
Dimensions	{oloc}/dict/dim/{version number}/dim.xsd	{ons}/dict/dim/{version-number}	{opre}_dim_{version-number}
Explicit domains	{oloc}/dict/dom/exp.xsd	{ons}/dict/exp	{opre}_exp
Typed domains	{oloc}/dict/dom/typ.xsd	{ons}/dict/typ	{opre}_typ

---

[12]   However, concepts that have never been used in production reporting may be deleted.

[13] Please note that version numbers used throughout this document (i.e. 3.4.0 for EBA and 2.9.0 in EIOPA) are just for illustration purposes and do not bind to any specific EBA or EIOPA releases.

Dictionary concept	Official location	Target namespace	Namespace prefix
Explicit domain members of domain	{oloc)/dict/dom/{dc}/mem.xsd	{ons)/dict/dom/{DC}	{opre}_{DC}

Examples of location, target namespace and its prefix for dictionary concepts are presented in the Table below:

Dictionary concept	Official location	Target namespace	Prefix
Metrics (except enumerated	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/met/met.xsd	http://www.eba.europa.eu/xbrl/crr/dict/met	eba_met
Metrics (enumerated)	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/met/3.4.0/met.xsd	http://www.eba.europa.eu/xbrl/crr/dict/met/3.4.0	eba_met_3.4.0
Dimensions	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dim/3.4.0/dim.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dim/3.4.0	eba_dim_3.4.0
Explicit domains	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/exp.xsd	http://www.eba.europa.eu/xbrl/crr/dict/exp	eba_exp
Typed domains	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/typ.xsd	http://www.eba.europa.eu/xbrl/crr/dict/typ	eba_typ
Explicit domain members (domain CP)	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/cp/cp.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/CP	eba_CP
Dimensions	http://eiopa.europa.eu/eu/xbrl/dict/dim/2.9.0/dim.xsd	http://eiopa.europa.eu/xbrl/dict/dim/2.9.0	eiopa_dim_2.9.0
Explicit domains	http://eiopa.europa.eu/eu/xbrl/dict/dom/exp.xsd	http://eiopa.europa.eu/xbrl/dict/exp	eiopa_exp
Typed domains	http://eiopa.europa.eu/eu/xbrl/dict/dom/typ.xsd	http://eiopa.europa.eu/xbrl/dict/typ	eiopa_typ
Explicit domain members example (domain CG)	http://eiopa.europa.eu/eu/xbrl/dict/dom/cg/mem.xsd	http://eiopa.europa.eu/xbrl/dict/dom/CG	eiopa_CG
MD version metrics	http://eiopa.europa.eu/eu/xbrl/dict/met/met.xsd	http://eiopa.europa.eu/xbrl/dict/met	eiopa_met

8.2 Metrics

In general, metrics define the nature of the measure to be performed by doing the following:

– indicating the data type, i.e. expected type of value that should be reported for a data point,

– determining the period type, i.e. whether a fact corresponding to a data point is reported for a single date (instant) or period of time (duration),

– expressing certain semantics.

In EIOPA taxonomies there is a different treatment of metrics between HD and MD (for more information, see associated *EIOPA DPM Documentation*). Neither version applies period type differentiation of metrics - in both versions, period type is set to instant[14] (in some cases the duration of a data point may be expressed using certain dimensional properties). Please note that XBRL representation contains only MD metrics.

Similarly, in the EBA reporting all the contexts in an instance document are expected to include an xbrli:period element with the same value: the reference period[15] in the case of metrics of duration type, or the end of the reference period (for metrics of instant type). The variations from this reference period in certain data points are expressed with the Reference Period (RF) dimension. This approach has been introduced in order to overcome the difficulty of defining time constraints for multiple periods in the table and definition linkbases.

Technically, metrics are represented in the taxonomy as XBRL primary items and defined in schema files named *met.xsd* (in *{oloc}/dict/met/* folder location) that reference label linkbase file *met-lab-{lang}.xml* (providing human readable labels as defined in the DPM; for representation in syntax see *6.2.1 Standard labels*) and definition linkbase file *met-def.xml* (defining XBRL Dimensions relationships that constraint using of metrics in reports[16]).

The code (*{name}*) for each metric is composed of three components:

– a letter that represents the data type in lowercase (for available options, see table below),

– a letter that represents the period type characteristics (*i* for instant and *d* for duration, which as explained above is always *i* in the EIOPA taxonomy),

---

[14] This approach has been introduced in order to overcome the difficulty of defining time constraints for multiple periods in the table, definition and XBRL Formula specification based linkbases.

[15] Reference period is defined as the period that starts at the beginning of the accounting year and ends at the reference date.

[16] In order to prevent from unrequested content in filings, all metrics are prohibited from being reported (in the dictionary) unless they are subsequently used in hypercubes of tables referenced from a module (see next sections of this document).

> – a number that corresponds to the numeric code in the model (no zero padding or predetermined length).

Model data type	XBRL data type	Local name codification letter	Reporting unit
Monetary (currency)	xbrli:monetaryItemType	m	Adequate currency using ISO 4217 codification (e.g.: iso4217:EUR)
Percent or Ratio	num:percentItemType	p	xbrli:pure
Decimal[17]	xbrli:decimalItemType	r	xbrli:pure
Integer	xbrli:integerItemType	i	xbrli:pure
Date	xbrli:dateItemType	d	No unit
Boolean (true/false or 0/1)	xbrli:booleanItemType	b	No unit
Text	xbrli:stringItemType / model:notEmptyString	s	No unit
Enumerated	enum:enumerationItemType	e	No unit
True	model:trueItemType (restriction of xbrli:booleanItemType to "true")	t	No unit
URI	xbrli:anyURIItemType	u	No unit

In the case of enumerated types, additional attributes apply as per Extensible Enumeration specification.

The id of the element (necessary for XLink locators) is composed like this:

> *{opre}_{name}*

Where *{opre}* represents the prefix of the base namespace of the owner of the base item and *{name}* represents the name described above. Some examples follow:

---

[17]   In EIOPA taxonomies there are a few cases where decimal metrics' codes start with letter *p* rather than *r*. They were used in preparatory phase reporting where the naming codification was different (both percent/ratio and other decimal items were using *p* code letter).

Owner	Data / period type	Code	Name	Id	Namespace	Prefix
EBA	Monetary / Instant	7	mi7	eba_mi7	http://www.eba.europa.eu/xbrl/crr/dict/met	eba_met
EBA	Text / Instant	7	si7	eba_si7	http://www.eba.europa.eu/xbrl/crr/dict/met	eba_met
EBA	Enumerated	5	ei5	eba_ei5	http://www.eba.europa.eu/xbrl/crr/dict/met/3.4.0	eba_met_3.4.0
EIOPA	Decimal	17	pi17	eiopa_pi17	http://eiopa.europa.eu/xbrl/dict/met	eiopa_met
EIOPA	Integer	31	ii31	eiopa_ii31	http://eiopa.europa.eu/xbrl/dict/met	eiopa_met
EIOPA	Monetary	43	mi43	eiopa_mi43	http://eiopa.europa.eu/xbrl/dict/met	eiopa_met
EIOPA	Explicit domain	17	ei17	eiopa_ei17	http://eiopa.europa.eu/xbrl/dict/met/2.9.0	eiopa_met_2.9.0

8.2.1 Enumerated metrics

The allowed values for an enumerated metric can change over time, leading to different versions. Each version uses the same name, but is placed in a different namespace (see the example above), making them different XBRL items. To express that these XBRL items represent metrics are related, special *arcroles* are used. Being assured of the relationship users can make an informed decision to combine the data from different versions or not.

Example:

Id	Namespace	Values
eba_ei5	http://www.eba.europa.eu/xbrl/crr/dict/met/3.4.0	eba_AB:x1, eba_AB:x2
eba_ei5	http://www.eba.europa.eu/xbrl/crr/dict/met/3.5.0	eba_AB:x1, eba_AB:x3, eba_AB:x4
eba_ei5	http://www.eba.europa.eu/xbrl/crr/dict/met/3.6.0	eba_AB:x1, eba_AB:x3, eba_AB:x4, eba_AB:x5

The enumerated metric is introduced in dictionary version 3.4.0. In version 3.5.0 the allowed value *eba_AB:x2* is split into *eba_AB:x3* and *eba_AB:x4*. In version 3.6.0 another value *eba_AB:x6* is added.

The following relationships are added to the definition linkbase to link a specific version to the previous version and to the initial version.

From	To	Arcrole id
eba_met_3.5.0:ei5	eba_met_3.4.0:ei5	EnumeratedMetricPreviousVersion
eba_met_3.5.0:ei5	eba_met_3.4.0:ei5	EnumeratedMetricInitialVersion
eba_met_3.6.0:ei5	eba_met_3.5.0:ei5	EnumeratedMetricPreviousVersion
eba_met_3.6.0:ei5	eba_met_3.4.0:ei5	EnumeratedMetricInitialVersion

8.3 Domains

Explicit domains are represented using XBRL abstract items of domain type (*model:explicitDomainType*) in the schema file ("*exp.xsd*") with namespace {*ons}/dict/exp* and prefix {*opre}_exp*.

Typed domains are represented as XML elements that are *not* in the substitution group of *xbrli:item*. These elements are defined in the schema file ("*typ.xsd*")[18] with namespace {*ons}/dict/typ* and prefix {*opre}_typ*.

Both schema files are placed in *{oloc}/dict/dom/* folder location.

The code (*{name}*) of each domain corresponds to its code in the model (which is a short sequence of uppercase letters, usually two).

Value of the *id* attribute of a domain (necessary for XLink locators) is composed according to the following pattern: *{opre}_{name}.*

Where {opre} represents the prefix of the base namespace of the owner of the domain and {name} represents the name described above. Some examples follow:

Owner	Code	Element Name	Type	Id	Namespace	Prefix
EBA	CO	CO	Explicit	eba_CO	http://www.eba.europa.eu/xbrl/crr/dict/exp	eba_exp
EBA	MI	MI	Typed	eba_MI	http://www.eba.europa.eu/xbrl/crr/dict/typ	eba_typ
EIOPA	BC	BC	Explicit	eiopa_BC	http://eiopa.europa.eu/xbrl/dict/exp	eiopa_exp
EIOPA	ID	ID	Typed	eiopa_ID	http://eiopa.europa.eu/xbrl/dict/typ	eiopa_typ

---

[18] Explicit domains are xbrli:items whereas typed domains are not. Because of this, labels for the former ones are defined using standard label links and labels for the latter using generic label links. As some tools in the market do not support a single file with two different extended links, these items have been split into two different schemas.

Though the namespace of explicit and typed domains is different, different local names should be used to avoid any confusion.

Domain schema files reference label linkbase files[19] *exp-lab-{lang}.xml* and *typ-lab-{lang}.xml* (providing human readable labels as defined in the DPM; for representation in syntax see *section on labels*).

8.3.1 Explicit domain members and hierarchies

The local name (*{name}*) of each explicit domain member corresponds to its numeric code in the DPM Dictionary which in general it starts with lowercase letter *x* (due to XML naming restrictions disallowing digit as the starting character) followed by a sequential number. If the concept represented has already a widely accepted standard codification, like ISO codes or NACE codes, the local name will match the existing codification (usually in lower case). More specifically, the following ISO codes are used:

-   ISO 4217: standard currency codes composed of three alphabetical characters

-   ISO 3166-1 alpha-2: standard country codes composed of two alphabetical characters,

-   NACE codes: NACE code list (without dots).

The default domain member of a domain (usually, but not necessarily, the one with code *x0*) is marked with an attribute: *model:isDefaultMember* = "true".

The id of explicit domain members follows the general rule:

> {opre}_{name}

The schema file that represents explicit members is placed in a folder with the name of its corresponding domain according to the following pattern: *{oloc}/dict/dom/{dc}* where *{dc}* is domain code in lowercase. The schema file for explicit domain members is called "mem.xsd" and its namespace is constructed based on the following pattern: *{ons}/dict/dom/{DC}* while prefix consist of *{opre}_{DC}* where *{DC}* is domain code in the uppercase.

Owner	Domain code	Domain members schema	Namespace	Prefix
EBA	CO	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/co/mem.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/CO	eba_CO
EBA	MI	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/mi/mem.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/MI	eba_MI

---

[19] Explicit domains are of *xbrli:item* substitution group whereas typed domains are not. Because of this, labels for the former ones are defined using standard label links and labels for the latter using generic label links.

EIOPA	CM	http://eiopa.europa.eu/eu/xbrl/dict/dom/cm/mem.xsd	http://eiopa.europa.eu/xbrl/dict/dom/CM	eiopa_CM
EIOPA	GA	http://eiopa.europa.eu/eu/xbrl/dict/dom/ga/mem.xsd	http://eiopa.europa.eu/xbrl/dict/dom/GA	eiopa_GA

This schema file references linkbases defining labels (*mem-lab-{lang}.xml*) for domain members (according to the DPM dictionary) and a definition linkbase file (*mem-def.xml*) where all members are connected to the domain item using *domain-member* arcrole of XBRL Dimensions.

Hierarchies are represented using XBRL extended link roles whose role is built following this pattern:

{ons}/role/dict/dom/{dom-code}/{version-number}/{hierarchy-code}

Where *{ons}* represents the namespace of the owner, *{dom-code}* represents the code of the domain, *{version-number}* represents the version number of the release and *{hierarchy-code}* the numeric code of the hierarchy. The id of these roles is composed following the pattern: *{opre}_{dom-code}{hierarchy-code}*.

Owner	Domain code	Hierarchy Code	Role	Id
EBA	MI	1	http://www.eba.europa.eu/xbrl/crr/role/dict/dom/3.4.0/MI/MI1	eba_AP1
EIOPA	CM	1	http://eiopa.europa.eu/xbrl/role/dict/dom/2.9.0/CM/CM1	eiopa_CM1
EIOPA	GA	4	http://eiopa.europa.eu/xbrl/role/dict/dom/2.9.0/GA/GA4	eiopa_GA4
EBA	CO	1	http://www.eba.europa.eu/xbrl/crr/role/dict/dom/3.4.0/CO/CO1	eba_CO1

The schema file that represents hierarchies (defining role types and referring to linkbases) is placed under a *{version-number}* folder which is under the folder with the name of its corresponding domain and it is called "hier.xsd". Its namespace is constructed based on the following pattern: *{ons}/dict/dom/{DC}/{version-number}/hier* while prefix consist of *{opre}_{DC}_h* where *{DC}* is domain code in the uppercase

Owner	Domain code	Hierarchies schema	Namespace	Prefix
EBA	CO	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/co/3.4.0/hier.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/CO/3.4.0/hier	eba_CO_h
EBA	MI	http://www.eba.europa.eu/eu/fr/xbrl/crr/dict/dom/mi/3.4.0/hier.xsd	http://www.eba.europa.eu/xbrl/crr/dict/dom/MI/3.4.0/hier	eba_MI_h
EIOPA	CM	http://eiopa.europa.eu/eu/xbrl/dict/dom/cm/2.9.0/hier.xsd	http://eiopa.europa.eu/xbrl/dict/dom/CM/2.9.0/hier	eiopa_CM_h
EIOPA	GA	http://eiopa.europa.eu/eu/xbrl/dict/dom/ga/2.9.0/hier.xsd	http://eiopa.europa.eu/xbrl/dict/dom/GA/2.9.0/hier	eiopa_GA_h

- These schema files refer to a linkbase file containing hierarchy specific labels for members (*hier-lab-mem-{lang}.xml*) and a definition linkbase (*hier-def.xml*), which enables the inclusion of the members of a hierarchy in dimensional combinations or applying them as enumerations for metrics (using *domain-member* relationships of XBRL Dimensions 1.0. and taking into account the *xbrldt:usable* attribute to identify "grouping" members).

The root member of the definition and presentation relationship networks is the domain item, as defined in the *exp.xsd* schema associated with the owner.

Some hierarchies of members are used to constraint the values of metrics with means of the XBRL Extensible Enumerations specification. In this case the labels applicable to members in a particular enumeration may differ from the standard labels of these members. This requirement is addressed by defining member labels (using standard generic label role) in an extended link role specific to a hierarchy. Examples of such cases are provided in *Table 5*.

Table 5. Examples of hierarchy specific labels.

Member QName	Standard label	Hierarchy specific label ELR	Hierarchy specific label
eiopa_CN:x1	Reported	http://eiopa.europa.eu/xbrl/role/dict/dom/2.9.0/CN/CN20	1 - Reported
eiopa_CN:x20	Not reported as no off-balance sheet items		2 - Not reported as no off-balance sheet items
eiopa_CN:x2	Not reported other reason		0 - Not reported other reason (in this case special justification is needed)

8.3.2 Typed domain values

Values of typed domains are neither listed as XBRL items with labels nor arranged in hierarchies. The content of typed domains is restricted by an XML data type constraint (as these domains, according to the XBRL Dimensions specification, are XML constructs).

In most cases, a typed domain would be represented by an XML element with a simple data type (e.g. *model:notEmptyString* or *xs:decimal*), though further restrictions are technically possible (also with means of business rules defined according to XBRL Formula specification).

Typed domains may be *nillable="true"* which means that they can be reported as *xsi:nil="true"* (and no value). This construct is used in reporting of optional open table columns modelled as typed dimensions.

8.4 Dimensions

The representation of dimension items in XBRL is defined in the XBRL Dimensions 1.0 specification.A dimension could have different domain in different release, the schema file defining dimension items is placed in the *{oloc}/dict/dim/{version_number}* folder and named *dim.xsd* with namespace *{ons}/dict/dim/{version_number}* and *{opre}_dim_{version_number}* prefix.

The local name of each dimension corresponds to its code in the model: a short sequence of capital case letters (usually two, but it is not limited to two letters).

The id of the element (necessary for XLink locators) is composed like base items:

> *{opre}_{name}*

Where *{opre}* represents the prefix of the base namespace of the owner of the dimension and *{name}* represents the name described above. Some examples follow:

Owner	Code	Name	Id	Namespace	Prefix
EBA	CP	CP	eba_CP	http://www.eba.europa.eu/xbrl/crr/dict/dim/{version_number}	eba_dim_{version_number}
EBA	MC	MC	eba_MC	http://www.eba.europa.eu/xbrl/crr/dict/dim/{version_number}	eba_dim_{version_number}
EIOPA	VL	VL	eiopa_VL	http://eiopa.europa.eu/xbrl/dict/dim/{version_number}	eiopa_dim_{version_number}
EIOPA	VG	VG	eiopa_CG	http://eiopa.europa.eu/xbrl/dict/dim/{version_number}	eiopa_dim_{version_number}

The schema file defining dimensions includes references to (a) label linkbase file(s) *dim-lab-{lang}.xml* and a definition linkbase whose file name is "*dim-def.xml*" and is placed in the same folder as the schema file. This linkbase includes the following information about explicit dimensions:

- Reference to the domain associated to the dimension by means of a dimension-domain relationship (with xbrldt:usable attribute equal to "false") pointing to a domain item defined in either the *exp.xsd* or *typ.xsd* schema file of any referenced or defined owner.
- Reference to the default member of that dimension by means of a dimension-default relationship. Note that though the model defines default members at domain level, the dimensions XBRL specification establishes this relationship at dimension level. Thus, each dimension using a domain with a default member must include this relationship.

These relationships are defined in an extended whose role is the standard one (http://www.xbrl.org/2003/role/link).

8.5   Compound items

A compound item is an item (explicit domain member) whose definition comprises from combination of semantics of two or more dimension and explicit domain member pairs defined in the dictionary.

A well- known example of a compound item is treasury bills. A treasury bill is a specific type of debt security: issued by a central government and with an original maturity of less than 1 year. Other types of investment types are treasury bonds (or T-Bonds) that have a maturity of 30 years and T-Notes that have maturity between 1 and 30 years.

The relationship between these explicit domain members is captured in the definition linkbase. A dedicated extended link role defined in the Eurofiling model schema (black box in the example below) is used to store relations describing the composition of compound items. XBRL Dimensions specification arcroles domain-member and dimension-domain are used to link respectively:

the contributing item (explicit domain member) with its explicit domain and

that explicit domain with the dimension providing context to the use of the explicit domain member in the compound item definition.

A custom arcrole defined in the Eurofiling model schema is applied to link these dimensions to the compound item (green box in the example below).

```
<link:definitionLink xlink:type="extended" xlink:role="http://www.eurofiling.info/eu/fr/xbrl/ext/compoundItemBreakdown">

  <link:loc xlink:type="locator" xlink:href="treasury-bills.xsd#citb_treasuryBill" xlink:label="loc_compound_item-t-bills"/>

  <link:loc xlink:type="locator" xlink:href="treasury-bills.xsd#citb_issuerSector" xlink:label="issuerSector" xlink:title="issuerSector"/>
  <link:definitionArc xlink:type="arc" xlink:arcrole="http://www.eurofiling.info/xbrl/arcrole/compoundItem-definition" xlink:from="loc_compound_item-t-bills" xlink:to=
"issuerSector" xlink:title="3" order="2.0"/>

  <link:loc xlink:type="locator" xlink:href="treasury-bills.xsd#citb_sectors" xlink:label="sectors" xlink:title="sectors"/>
  <link:definitionArc xlink:type="arc" xlink:arcrole="http://xbrl.org/int/dim/arcrole/dimension-domain" xlink:from="issuerSector" xlink:to="sectors" xlink:title="4" order="1.0"/>

  <link:loc xlink:type="locator" xlink:href="treasury-bills.xsd#citb_centralGovernment" xlink:label="centralGovernment" xlink:title="centralGovernment"/>
  <link:definitionArc xlink:type="arc" xlink:arcrole="http://xbrl.org/int/dim/arcrole/domain-member" xlink:from="sectors" xlink:to="centralGovernment" xlink:title="5" order="1.0"/>
```

9 Reporting requirements layer

Frameworks, tables, modules and other concepts constitute the layer of the model where actual reporting requirements are specified with the support of the financial concepts defined in the dictionary.

All the files that correspond to this layer are placed under the folder "*fws*" in the official location of its owner (i.e. *{ons}/fws/*). Its namespace is obtained by adding the suffix "*fws*" to the base namespace of the owner plus some additional suffixes that depend on the type of concept represented.

Note: in EIOPA XBRL taxonomies, frameworks are defined for the MD modelling approach only.

9.1 Frameworks and their releases
Frameworks are public elements represented using XBRL abstract items of framework type ("*model:frameworkType*") in the schema file "*fws.xsd*".

Schema property	Value
Official location	{oloc}/fws/fws.xsd
Target namespace	{ons}/fws
Target namespace prefix[20]	{opre}_fws
Element local name	{framework}
Element id	{opre}_{framework}

---

[20] Target namespace prefixes are not strictly necessary. Moreover, schemas like frameworks define names that are not used in the exchange of information between supervisors and supervised entities. However, as some XBRL tools raise warnings whenever they find a schema with no prefix defined. So, prefixes have been included to avoid misleading the users of these tools.

The local name of each framework element corresponds to its code in the model (*{name}*) and its *id* follows a general pattern (*{opre}_{name}*). Examples of frameworks are presented in table below:

Owner	Schema property	Value
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/fws/fws.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/fws
	Target namespace prefix	eiopa_fws
	Local name example	s2, pf, pepp
	Element id example	eiopa_s2
	Element label (English) example	Solvency II (MD version)
EBA	Official location	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/fws.xsd
	Target namespace	http://www.eba.europa.eu/xbrl/crr/fws
	Target namespace prefix	eba_fws
	Local name example	finrep, corep, ae
	Element id example	eba_finrep, eba_corep, eba_ae
	Element label (English) example	FINREP

Each framework has a folder where the files of its releases are placed. This folder has the name of its code in the model:

Description	Framework folder
Common Reporting	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/
Financial Reporting	http://www.eba.europa.eu/eu/fr/xbrl /crr/fws/finrep/
Asset Encumbrance	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/ae/
Solvency II	http://eiopa.europa.eu/eu/xbrl/fws/s2/
Pension funds	http://eiopa.europa.eu/eu/xbrl/fws/pf/

Each framework has releases, as the underlying legislation is updated or for technical purposes. A particular release of a framework is called a taxonomy and is identified by the code of the framework followed by the version number: *{oloc}/fws/{framework}/{version-number}*. A taxonomy contains the modules, tables and validation rules that are added or updated in the particular release.

Fictional examples

Description	Release Version	Taxonomy folder

Common Reporting	3.4.0 released on 15 Nov, 2020	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/3.4.0/
Financial Reporting	3.4.0 released on 1 Jun 2022	http://www.eba.europa.eu/eu/fr/xbrl /crr/fws/finrep/3.4.0/
Solvency II	2.9.0 released on June 15, 2026	http://eiopa.europa.eu/eu/xbrl/fws/s2/2.9.0/
Pension Funds	2.9.0 released on June 15, 2026	http://eiopa.europa.eu/eu/xbrl/fws/pf/2.9.0/

The folder of a taxonomy includes at least three folders for tables (*tab*), modules (*mod*) and validations (*val*).

9.2 Tables

The table folder includes a schema file (*tab.xsd*), The schema includes the definition of table groups (if any, e.g. template variants), which are represented using XBRL abstract items of table group type ("*model:tableGroupType*"). The name (*{name}*) of a table group item composed by adding the prefix *tg* to the code (*{table group code}*) of a table group in the model

Schema property	Value
Official location	{oloc}/fws/{framework}/{version-number}/tab/tab.xsd
Target namespace	{ons}/fws/{framework}/{version-number}/tab
Target namespace prefix	{opre}_tab
Element local name	tg{table-group-code}
Element id	{opre}_{local-name}

Examples in EBA/EIOPA taxonomy:

Owner	Schema property	Value
EBA	Official location	https://www.eba.europa.eu/eu/fr/xbrl/crr/fws/finrep/3.4.0/tab/tab.xsd
	Target namespace	http://www.eba.europa.eu/xbrl/crr/fws/finrep/3.4.0/tab
	Target namespace prefix	eba_tab
	Local name example	tgF_01.01
	Element id example	eba_tgF_01.01
	Element label (English)	Balance Sheet Statement [Statement of Financial Position]: Assets
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/fws/s2/2.9.0/tab/tab.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/fws/s2/2.9.0/tab
	Target namespace prefix	eiopa_tab

Local name example	tgS.01.01.01	
Element id example	eiopa_tgS.01.01.01	
Element label (English)	S.01.01.01 Appendix I: Quantitative reporting templates	

The files that define the content of each table are placed in a folder whose name corresponds to the code of the table in the model (*{table code}*) in lowercase.

Schema property	Value
Official location	{oloc}/fws/{framework}/{version-number}/tab/{table}/{table}.xsd
Target namespace	{ons}/fws/{framework}/{version-number}/tab/{table}
Target namespace prefix	{opre}_tab_{table}
Element local name	N/A (elements defined as resources in linkbases)
Element id	N/A

Examples in EBA/EIOPA taxonomy:

Owner	Schema property	Value
EBA	Official location	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/finrep/3.4.0/tab/f_01.01/f_01.01.xsd
	Target namespace	http://www.eba.europa.eu/xbrl/crr/fws/finrep/its-005-2020/3.4.0/tab/F_01.01
	Target namespace prefix	eba_tab_F_01.01
	Local name example	N/A
	Element id example	eba_tF_01.01 (table resource id in the table linkbase)
	Element label (English)	F 01.01
	Table folder	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/finrep/3.4.0/tab/f_01.01/
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/fws/s2/2.9.0/tab/ s.01.01.01.01/s.01.01.01.01.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/fws/s2/2.9.0/tab/S.01.01.01.01
	Target namespace prefix	eiopa_tab_S.01.01.01.01
	Local name example	N/A
	Element id example	eiopa_tS.01.01.01.01 (table resource id in the table linkbase)
	Element label (English)	S.01.01.01.01
	Table folder	http://eiopa.europa.eu/eu/xbrl/fws/s2/2.9.0/tab/ s.01.01.01.01/

A schema file for a table refers to:
   a.   a table linkbase (*{table code}-rend.xml*),

   b.   a definition linkbase (*{table code}-def.xml*),

   c.   a generic label linkbase with table texts (*{table code}-lab-{lang}.xml*),

d.   a generic label linkbase with table codes (*{table code}-lab-codes.xml*),

e.   If applicable, a generic label linkbase file identifying types of key columns in case of open tables (*{table code}-lab-keys.xml*),

The table linkbase (*a*) includes the definition of the table according to the Table Linkbase specification. The relationships of each table are placed in an extended link whose role is built according to the following pattern: *{ons}/role/fws/{framework}/{version-number}/tab/{table code}* with id *role*. For example, table linkbase relationships for EIOPA table S.01.01.01.01 are defined in extended link role *http://eiopa.europa.eu/xbrl/role/fws/solvency/solvency2/2.9.0/tab/S.01.01.01.01*.

In this linkbase, the different components of tables are represented using resources. The "id" of these resources is based on the code of the model plus a prefix to obtain a unique code in the context of the linkbase file:

Resource	Id pattern	Example
table:table	{opre}_t{table code (uppercase)}	eiopa_tS.01.01.01.01, eba_tC_84.00.w
table:breakdown (predefined or variable axis)	{opre}_a{sequential number}	eiopa_a1 eba_a1
table:conceptRelationshipNode; table:dimensionRelationshipNode	{opre}_r{sequential number}	eiopa_r6
top level abstract table:ruleNode (ex: aspectNode)	{opre}_a{sequential number}.root	eiopa_a1.root eba_a1.root
table:ruleNode	{opre}_c{sequential number}	eiopa_c2 eba_c1
filer, e.g. df:explicitDimension	{opre}_a{sequential number}.root.filter	eiopa_a3.root.filter eba_a3.root.filter

According to the table specification, aspect rules are used to specify the concepts represented in predefined axes.

Although not strictly requested by the Table Linkbase specification, *link:roleRef* is included in the table linkbase files pointing to an extended link role when resources relate to domain member relationships defined in the dictionary.

The definition linkbase file (b) includes dimensional relationships valid in the context of the table. Valid combinations are defined using only positive (all) closed hypercubes obtained from the set of valid cells of the table following an optimization algorithm[21].

Each extended link role contains a set of primary items (metrics) and a single hypercube[22]. In case of multiple primary items, the first one will be used to group the rest and reduce the number of "all" arcs. The domain element will be used as target of dimension-domain arcs to avoid cycles. The @xbrldt:targetRole attribute might be necessary in the case of hypercubes with dimensions sharing the same domain.

The roles of the extended links necessary to express these combinations are built adding numeric suffixes to the role previously defined for the table. For example:

> *{ons}/role/fws/{framework}/{version-number}/tab/{table}/1*

> *{ons}/role/fws/{framework}/{version-number}/tab/{table}/2*

> ...

The generic label linkbase file of a table contains labels for Table Linkbase nodes. In addition to the standard label, a filing-indicator-code label also contains a documentation label which defines a code to be used on filing indicators (see next section of this document).

Another (separate) generic label linkbase file (*d*) referenced from table schema file contains codes. These are row/column/sheet/table codes (*http://www.eurofiling.info/xbrl/role/rc-code*) for table rule nodes (e.g. *C0010*, *R0070, S.01.01.01.01*) and a filing indicator code (using the *http://www.eurofiling.info/xbrl/role/filing-indicator-code* role) for the table resource identifying a value to be included on a filing indicator when a template (which a table is part of) is reported or explicitly not reported (e.g. *S.02.01*, see section on filing indicators)).

Open tables in e.g. EIOPA taxonomies may contain an optional generic label linkbase file (*e*) identifying key column types using *http://eiopa.europa.eu/eu/xbrl/role/key-column-type* role.

9.3 Modules

Modules serve as entry points to subsets of information requirements that shall be used for filing (the only files referenced from XBRL instance documents) depending on the

---

[21] It is important to remark that XBRL hypercubes in the definition linkbase of tables are validation artefacts and should not be used by external systems for the automatic creation of database structures. The hypercubes produced by the algorithm do not obey to any kind of business criteria. These hypercubes might be modified with the addition of new information to tables with the only purpose of reducing the final set of hypercubes and performing more efficiently with XBRL market tools.

[22] The model schema includes a hypercube element to be used. There is no need to define hypercube elements in each table or taxonomy.

reporting scenario (reporting frequency, solo or group data, etc.) as defined in the underlying model.

Modules are represented using XBRL abstract items of module type ("*model:moduleType*").

Module elements include two optional attributes that establish its currency period: the starting date of the period interval (*model:fromDate* attribute) and its end date (*model:toDate* attribute). In general, the "fromDate" attribute should always be included: it indicates the starting reference date of this module version. If the "toDate" attribute is not included, then the element is assumed to be current for any period after the "fromDate" attribute until fromDate -1 of the new version of this same module.

Each module is stored in a different schema file whose module file name is the same as the code of the module in the model plus the extension ".xsd". These schema files import the schemas of all the tables imported by that module:

Schema property	Value
Official location	{oloc}/fws/{framework}/{version-number}/mod/{module}.xsd
Target namespace	{ons}/fws/{framework}/{version-number}/mod/{module}
Target namespace prefix	{opre}_mod_{module}
Element local name	{module}
Element id	{opre}_{module}

Examples in EBA/EIOPA taxonomies

Owner	Schema property	Value
EBA	Official location	http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/finrep/3.4.0/mod/ifrs9.xsd
	Target namespace	http://www.eba.europa.eu/xbrl/crr/fws/finrep/its-005-2020/3.4.0/mod/FINREP9
	Target namespace prefix	eba_mod_FINREP9
	Local name example	FINREP9
	Element id example	eba_FINREP9
	Element label (English)	Finrep Reporting (IFRS9)
EIOPA MD version	Official location	http://eiopa.europa.eu/eu/xbrl/fws/s2/2.9.0/mod/ars.xsd
	Target namespace	http://eiopa.europa.eu/xbrl/fws/s2/2.9.0/mod/ars
	Target namespace prefix	eiopa_mod_ars
	Local name example	ars
	Element id example	eiopa_ars

Module schema files import the schemas of all the tables required by that module (from the *tab* folder of the taxonomy; determining the subset of information requirements for a particular reporting scenario defined by a module). They also import the filing indicators schema file and a schema file referring to custom functions' definition and implementation.

In addition to these imports, the module schema file references also a number of linkbase files:

- label linkbase file(s) with label for a module (*{module code}-lab-{lang}.xml*),

- presentation linkbase (*{module code}-pre.xml*) where the relationships between modules, table groups and tables are expressed using the legacy *group-table* arcs (defined in the Eurofiling *model.xsd* schema file),

- a general linkbase *{module code}-val.xsd* in val folder which references to

   o a linkbase file defining precondition tests for filing indicators (*{module code}-find-prec.xml*),

   o a linkbase file defining the applied tables for the assertions in a module ({module code}-val-tabs.xml)

   o optional value assertion definition (*{module code)-find-check.xml*), label (*{module code)-find-check-lab-{lang).xml*) and error message (*{module code)-find-check-err-{lang).xml*) checking and documenting the values of filing indicators applicable to a module,

   o optional severity level information of validation rules *{module code}-val-severity.xml* in the set folder of the taxonomy in scope.

   o value assertions (validation rules) definitions, labels and error messages declared in the *val* folder of the taxonomy in scope that is applicable to the tables imported by a module,

   o supportive constructs that may be used in defining XBRL Formula assertions (from the *http://*www.*eurofiling.info/eu/fr/xbrl/val* folder),

- in case of EIOPA - a linkbase file that is used to deactivate assertions as described in https://eurofiling.info/portal/taxonomiesmechxml-blacklist/.

Here are some examples of modules:

Module	Description

corep_of	Common reporting own funds
corep_le	Common reporting large exposures
corep_lcr	Common reporting liquidity coverage ratio
corep_nsfr	Common reporting net stable funding ratio
finrep	Financial reporting
ae	Asset Encumbrance

In EBA taxonomies, some of the modules contain a general information table "00.01" that must be included with any XBRL report. This provides general information describing the nature of the report (i.e. consolidation status and accounting standard).

These modules contain validation rules restricting the descriptive values of table 00.01 to appropriate values.

9.4   Filing indicators

The principle of proportionality stipulates that an entity's reporting burden should be proportional to its size. It allows a filer to report less information if it satisfies certain criteria. For example, this principle allows a smaller organisation to file less information if it is not active in some domains or if some figures are under a given threshold.

The evident technical solution to this business requirement would be to define a module (an entry point) for each reporting scenario. Each entry point would then only contain the subset of the model and validation checks specific to the reporting scenario in question. However, if several characteristics and/or thresholds are defined to cope with the proportionality principle, a different entry point must be defined for each and every valid combination of characteristics. This complicates:
– the filing process, where the filer must choose the appropriate entry point from

a potentially large selection which differ in subtle ways,

– the taxonomy, where several entry points must be defined, tested and assured

with added complexity if some assertions are shared between entry points and

some are not (which is typically the case),

– the submission handling process, where the received instances must be

processed against one of many different entry points,

– the maintenance of the taxonomy, where every time a new characteristic or

threshold is introduced for proportionality, the number of entry points could be

as much as doubled.

To overcome these difficulties the principle of "filing indicator" was introduced.

The idea of a filing indicator enables entry points to be shared between different similar reporting scenarios. The content of each entry point is notionally split into several components and every component (typically corresponding to a template) which is reported in an instance is accompanied by an explicit indication that the reporting unit has been filed.

Filing indicators serve the purpose of communicating the scope of the reported data based on templates. The main purposes of filing indicators are to:

- provide hints to applications using the taxonomy, when processing instance files, on which templates are included in the filing and, for example, shall be displayed to users,

- trigger execution of business rules (XBRL assertions) to be ran on a filing to check its correctness depending on the reported scope of data.

In technical terms, filing indicators are facts included as part of an instance document where the filer provides information about the reported templates (within the scope defined by a module that the filing is defined against, see previous section on Modules).

For traditional XBRL reporting documents, the elements and attributes used to communicate filing information are defined in the namespace *http://www.eurofiling.info/xbrl/ext/filing-indicators*. The official location of this schema file is https://www.eurofiling.info/eu/fr/xbrl/ext/filing-indicators.xsd. This schema file is imported in every taxonomy module (in case of EIOPA taxonomies this is done through its EIOPA counterpart - *http://eiopa.europa.eu/eu/xbrl/ext/filing-indicators.xsd*). Throughout this document, the prefix "find" will be used to make reference to this schema namespace.

For xBRL-CSV reporting documents, the elements and attributes used to communicate filing information are defined in the namespace *http://www.xbrl.org/taxonomy/int/filing-indicators/REC/2021-02-03*. The official location of this schema file is https://www.xbrl.org/taxonomy/int/filing-indicators/REC/2021-02-03/filing-indicators.xsd. This schema file is imported in every taxonomy module. Throughout this document, the prefix "fi" will be used to make reference to this schema namespace.

The following instance excerpt represents a filing with information about template with code C_01.00 and no information (explicitly stated) on template C_07.00:

For a traditional XBRL reporting document:

```
<find:fIndicators>
  <find:filingIndicator contextRef="ctx">C_01.00</find:filingIndicator>
  <find:filingIndicator contextRef="ctx" filed="false">C_07.00</find:filingIndicator>
</find:fIndicators>
```

Contexts to which facts representing find:filingIndicator element refer must identify the reporting entity and use the end date of the reporting period as the instant date.

For an xBRL-CSV reporting document:

```
templateID,reported
C_00.01,true
C_07.00,false
```

Filing rules determine how filing indicators must be provided in the XBRL report.

Identification of templates on find:filingIndicator facts (and on TemplateID) is made using codes. These codes are represented as label resources with the following role, as defined in the model schema:

http://www.eurofiling.info/xbrl/role/filing-indicator-code

These code labels are applied to either a table:table resource (in case a template is reflected by a single individual table) or to each of a set of tables that collectively represent a template. If one or more tables that are part of a template are reported, the corresponding filing indicator should be set (but at most one filing indicator of any code is needed).

Note: EIOPA information requirements include a *Content template* for each module that detail which templates have been included in a filing and why occasionally a template has been omitted. Filing indicators may in addition appear to serve the same purpose as the content templates but filing indicators are a technical mechanism (using XBRL tuples to distinguish from other facts and simplify referring from preconditions on assertions) which has been used to align with the EBA and the content templates satisfy a business requirement for reasoning behind the inclusion (or not) of templates in a report. There are a series of assertions which ensure that entries in the content table and values of filing indicators are consistent.

Additionally, validation rules associated with each module check if the reported values of filing indicators match the required codes. This rule is defined as an assertion in the *{module code}-find-check.xml* file. The test expression is for example *$filingIndicator = ('S.01.01','S.02.01')*. Documentation (label) and error message for this check is defined in *{module code}-find-check-lab-{lang}.xml* and *{module code}-find-check-err-{lang}.xml* files respectively.

In order to block the use of *xbrli:scenario* and *xbrli:segment* on contexts that filing indicator elements refer to, EIOPA extended the Eurofiling schema defining filing indicators with a definition linkbase where filing indicators are linked to a closed hypercube with no dimensions attached (files *filing-indicators.xsd* and *filing-indicators-*

def.xml in *http://eiopa.europa.eu/eu/xbrl/ext/* folder). Additionally, this folder includes also two XBRL assertions (*filing-indicators-check.xml*, *filing-indicators-check-lab-en.xml*, *filing-indicators-check-err-en.xml*):

- existence assertion (*filingIndicatorsExistanceAssertion*) checking if filing indicators are present in a report,

- value assertion (*filingIndicatorOutsidefIndicatorsTupleAssertion*) checking if filing indicator elements (*find:filingIndicator*) are declared in *find:fIndicators* tuple.

9.5 Validation rules

Data checks are created according to the XBRL Formula Specification 1.0.

9.5.1 Assertions

Validations are expressed using XBRL assertions, where each validation rule is implemented in one or more assertions. Assertions are being identified by a unique code, which is the code used to identify the corresponding validation rule expressed in the ITS documentation suffixed by _1, _2 etc.

For example, for the validation rule eba_v2285_h, its assertions are coded v2285_1_h, v2285_2_h etc.

An assertion is stored in a linkbase file with name starting with prefix *vr-* followed by the *{validation rule code}* and *.xml* extension (for example *vr-bv38-1.xml*). Each assertion is associated in the same linkbase file with a label, explaining the validation rule in business/form-centric terms, and an error message (according to the Generic messages 1.0 specification), which should be used by tools in case of an unsatisfied evaluation. Both the generic label and message resources may use different roles to identify different type of documentation or notes.

Depending on the impact on performance of size of the linkbase file, one or more assertions may be stored in the same linkbase.

In most cases, "value" assertions are used to express validation rules; for potential exceptions see *section9.5.5*. They refer to variables (usually fact variables however the use of generic variables is also allowed) which are named after the letters of the alphabet (i.e. *a, b, c, d*, …). Both the assertions and fact variables may refer to filters that can be complemented if necessary. Variables may bind as sequence and contain fallback values.

Assertion resource *xlink:label* and *id* attributes follow a pattern e.g. *{opre}_{RULE CODE}* (e.g. *eiopa_BV38-1*). *Xlink:label* and *id* of variable resources are constructed by adding a variable name to the assertion code, i.e. *{opre}_{RULE CODE}.{variable name}* (e.g. *eiopa_BV38-1.a*). Filter resources *xlink:label* and *id* are constructed according to the pattern: *{opre}_{rule code}.f{sequential number}* (e.g. *eiopa_BV38-1.f1*).

Not all assertions are applicable to every module. Each module includes the assertions that it needs.

Each assertion may also, in future taxonomies, be associated to two attributes: *model:fromDate* and *model:toDate* which may be used to express a period of validity, in which the reporting reference date should fall for the assertion to be evaluated.

9.5.2 EBA assertion patterns

In EBA there are several common patterns of validations implemented in the taxonomy, explained hereafter, which are:

- Hierarchy checks (Dimensional aggregation)

- Sign checks

- "Manual" or general value checks

9.5.2.1 Hierarchy checks (Dimensional Aggregation)

Derived from information in the data point model, the Hierarchy check (dimensional aggregation and metric aggregation) pattern corresponds to the validation of an aggregation of a business concept, or a set of business concepts, along a dimension. In other words, the rolling up of component parts of a breakdown along a particular aspect.

These rules have the suffix "_h", e.g. v0150_h. This rule, expressed in the ITS as "Table: C_02.00, Column: 010, Formula: {r490} = +{r500} + {r510}", is derived from the hierarchy with code PL2, which indicates a (fairly obvious) relationship between three possible values for the Portfolio dimension:

```
Banking and trading book =   Banking book
                           + Trading book
```

These three different values for the Portfolio dimension are the distinguishing factor of rows 490, 500 and 510 on table C_02.00, so this validation rule asserts that these rows should be related in the way the hierarchy indicates.

9.5.2.2 Sign checks

Many cells (data points) to be reported are required to be positive numbers or amounts (and conversely many are required to be negative). Where this is the case, this is enforced using sign check assertions, with the suffix "_s", which are also derived from information in the DPM.

E.g. v2468_s checks whether the values in column 050 and rows 010, 020 and 090 of table C 05.02 are negative (or zero).

Note that where a range of both rows and columns are checked for a particular sign, the table centric formula of these rules may initially appear strange, e.g. v2028_s "F_46.00

(r010;040;210, c090;110) : {F_46.00} <=0". This does not indicate, as the formula might suggest at first glance, that the table as whole is somehow less than or equal to zero, but that the (six) cells at the intersections of rows 010,040 and 210 and columns 090 and 110 must be.

9.5.2.3 "Manual" or general value checks

Moving beyond the information captured in a structured form in the DPM, and the validation rules that can be inferred from it, there are many additional business checks between data points. These have been specified individually by subject matter experts, have the suffix "_m", and involve a wide variety of formulae, e.g. v0219_m "{C_03.00, r020,c010} = {C_01.00, r020,c010} – {C_02.00, r010,c010} * 4.5%", or v0284_m "{C_06.00, c180} >= {C_06.00, c200}"[23].

9.5.3 Assertion and its applicable tables

Each assertion could be associated to a table (or tables[1]) it applies to, the link between an assertion and the table(s) it applies to is represented using applies-to-table arcs from the assertion to the resource that corresponds to the table. The URI of this arc is http://www.eurofiling.info/xbrl/arcrole/applies-to-table

Ex.#	Assertion example (textual description)	Assertion name	Tables
1	$a > 0 (where $a represents data in table 1)	vr1	table1
2	$a > 0 (where $a represents data in tables 1, 2 and 3)	vr1	table1
			table 2
			table 3
3	$a = $b (where $a represents data in table 1 whereas $b represents data in table 2)	vr1	table 1
			table 2

This table application information is defined for each module in a *val* folder file *{module code}-val-tabs.xml* from where is referred to respective assertions in the *val* folder, for example:

---

[23] Or even v1037_m "sum({F 31.01, r120, (c010-050)}) <= {F 10.00, r290,c030} - sum({F 10.00, c030, (r050-060, r110-120, r170-180)}) + {F 11.01, r500,c030} - sum({F 11.01, c030, (r040-050, r090-100, r140-150, r270-280, r320-330, r370-380)}) + {F 11.02, r230,c010} - sum({F 11.02, c010, (r040-050, r090-100, r140-150)})" !

```
<link:loc xlink:type="locator" xlink:href="../tab/c_106.00/c_106.00-rend.xml#eba_tC_106.00" xlink:label="loc_eba_tC_106.00" />

<link:loc xlink:type="locator" xlink:href="vr-v4189_a.xml#eba_v4189_a" xlink:label="loc_eba_v4189_a" />

<gen:arc xlink:type="arc" xlink:arcrole="http://www.eurofiling.info/xbrl/arcrole/applies-to-table" xlink:from="loc_eba_v4189_a" xlink:to="loc_eba_tC_106.00" />
```

9.5.4 Preconditions and filing indicator parameters

Each value assertion is associated to a precondition[24] on filing indicators. To avoid XBRL instance syntactic dependencies (e.g. using an Xpath expression), preconditions include a reference to a filing indicator parameter (no variableset-variable arc are required). The value of this parameter is set by a call to the XBRL int. registered function that checks for a positive filing indicator[25] in the instance document. This way, there is no need to provide externally a value to the processor (the value from the instance is used), the parameter is guaranteed to be only evaluated once (providing more chances for processors to perform optimizations), precondition expressions are simpler, and it makes possible, for more advanced uses, to override this value at application level (for instance, if the filing requirements of a credit institution are known, an application could override the values for filing indicator parameters rather than accepting the values provided by the filter).

A filing indicator parameter is defined for each table defined in the framework. These parameters are defined in the namespace of the filing indicators schema and have a name according to the following convention:

t{table-code}

where table-code represents the code of the corresponding table. Thus, the definition of one of these parameters would look like this:

```
<variable:parameter
    name="find:t{table-code}"
    select="xfi:positive-filing-indicator('template-co]de') ""
    as="xs:boolean" …/>
```

Where 'template-code' represents the code of the template

Each precondition is composed as a sequence of expressions that correspond to each set of tables where the validation is to be applied. Depending on the case, a combination of *or-* and and-expressions may be constructed:

```
"$find:t{c1.1} and $find:t{c1.2} and …

or $find:t{2.1} and $find:t{2.2} and …
```

---

[24] Assertions might have additional preconditions as required by the logic of the assertion to be tested. But these additional preconditions do not depend on filing indicators.
[25] xfi:positive-filing-indicator (xbrl.org)

or …"

Some examples:

Expression	Explanation
$find:t1	Assertion applies only to table 1. It will be evaluated if table 1 is marked as reported.
$find:t1 and $find:t2	Assertion crosses information between tables 1 and 2. It will be evaluated if table 1 and table 2 are marked as reported.
$find:t1 or $find:t2	Assertion applies to both table 1 and table 2, but considered in an individual way (there are no cross checks). It will be evaluated if table 1 or table 2 or both are reported.
$find:t1 and $find:t2 or $find:t3 and $find:t4	Assertion performs cross-checks between information in table 1 and table 2 on the one hand. On the other hand, it cross-checks information between table 3 and 4. It will be evaluated if table 1 and table 2 is reported or if table 3 and table 4 is reported or when all tables (1, 2, 3 and 4) are reported.

These preconditions are defined for each module in the val folder file *{module code}-find-prec.xml* from where they refer to respective assertions in the *val* folder, for example:

```
<variable:precondition xlink:type="resource" xlink:label="findPrec" test="$find:tS.03.01" />
<link:loc xlink:type="locator" xlink:href=".lvr-138.xml#eiopa_138" xlink:label="loc_eiopa_138" />
<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
xlink:from="loc_eiopa_138" xlink:to="findPrec" />
```

9.5.5 Existence assertions

Existence assertions are not compatible with the precondition-based control schema proposed in the previous chapter. Existence assertions perform a test on the number of evaluations of a set of variables. Preconditions restrict the number of evaluations of the assertion, but not the evaluation of the assertion itself. Consequently, existence assertions are always evaluated (unless controlled using assertion sets); if a filing indicator precondition is added to an existence assertion, it will raise false errors.

Existence assertions currently are only used to check the existence facts of 00.01 table. Ex: count({A 00.01, r0020, c0010}) > 0

In EBA/EIOPA taxonomies existence assertions are re-defined as value assertions using in addition the "pivot variable" – a fact variable that matches data in the instance document known to be reported always (it is defined once as a sequence variable that matches the filing indicators and uses aspect cover filters to avoid any interference with other variables). The rest of variables in the original existence assertion are included

with a fallback value (a value given to the variable if the fact is not found in the instance document).

The pivot-variable is defined in the namespace *http://www.eurofiling.info/xbrl/ext/pivot-variable*. The official location of this schema file is *https://www.eurofiling.info/eu/fr/xbrl/ext/pivot-variable.xsd*.

Though unlikely, there might be the case of validations that cannot be (effectively or efficiently) defined using value assertions. If such rules were required, the *id* attribute value of such assertions would follow a predefined naming convention (to be established when such situation occurs) to help applications not relying on validation sets to discard such evaluations.

9.5.6 Assertions severity

Each assertion is assigned with a severity as defined in the XBRL Assertions Severity specification (19 April 2016).

A locator for the severity resource and an arc linking it to an assertion is defined in file *{module code}-val.severity.xml* in the set folder alongside the val folder.

Currently the taxonomy applies ERROR and WARNING severity levels. In case of ERROR the submission is blocked. WARNING does not block the request, but allows to provide information about possible discrepancies.

In EIOPA there are specific concepts in the Basic Information template that may impact severity of assertions. In particular for non-regular reporting all assertions with ERROR severity levels are downgraded to WARNING. In order to achieve that these assertions are duplicated in the taxonomy with a suffix "_W" e.g. "vr-bv6-1.xml" with ERROR severity has a counterpart "vr-bv6-1_w.xml" with severity WARNING. See the next section for details.

It is considered to replace this mechanism with a dynamic assertion severity as per recently published XBRL specification: Assertion Severity 2.0 (xbrl.org).

9.5.7 Preconditions for specific concept's parameters (EIOPA only)

There are specific concepts included in the taxonomy (in particular in the basic information template) that identify if a submission is under a regular or non-regular reporting scenario. Similar to filing indicators, these facts taking specified values are declared on parameters and used on preconditions in order to trigger rules with different severity (ERROR or WARNING). These are also declared in *find-params.xml*

```
<variable:parameter xlink:type="resource" xlink:label="regularReporting" name="regularReporting"
select="eiopa_met:ei1677 = xs:Qname('eiopa_CS:x35') or eiopa_met:ei2503 = xs:Qname('eiopa_CS:x35')"
as="xs:boolean" id=" regularReporting" />
<variable:parameter xlink:type="resource" xlink:label="nonRegularReporting" name="nonRegularReporting"
select="eiopa_met:ei1677 != xs:Qname('eiopa_CS:x35') or eiopa_met:ei2503 != xs:Qname('eiopa_CS:x35')"
as="xs:boolean" id="nonRegularReporting" />
```

and subsequently applied in *{module code}-find-prec.xml* files:

```
<variable:precondition xlink:type="resource" xlink:label="rp" test="$regularReporting" />    <variable:precondition
xlink:type="resource" xlink:label="nrp" test="$nonRegularReporting" />
<link:loc xlink:type="locator" xlink:href=".vr-bv6-1.xml#eiopa_BV6-1" xlink:label="loc_eiopa_BV6-1" />
<link:loc xlink:type="locator" xlink:href=".vr-bv6-1_w.xml#eiopa_BV6-1_W" xlink:label="loc_eiopa_BV6-1_W" />
<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
xlink:from="loc_eiopa_BV6-1" xlink:to="rp" />
<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
xlink:from="loc_eiopa_BV6-1_W" xlink:to="nrp" />
```

9.5.8 Interval Arithmetic

An arithmetic comparison may not be exact due to rounding of figures and their representation. For example in a simple expression *A = B / C* where *B = 1000*, *C =3000* the result of division is *0.333333*…. If *A* is reported as *0.33* then compared to the result would raise an error and the rule can never be satisfied.

In order to handle the error margin caused by the imprecision of input data, assertions make use of a set of functions implemented according to the Custom Functions Implementation specification. These functions use the same name as the ones defined in the XPath 2.0 Functions specifications, but are defined in the following namespace *http://www.eurofiling.info/xbrl/func/interval-arithmetics* (with canonical prefix *iaf*) and placed in the location: *https://www.eurofiling.info/eu/fr/xbrl/func/interval-arithmetics.xml*. An entry point (referred from taxonomy modules) for these functions and additional ones that could be provided in the future is placed in the *https://www.eurofiling.info/eu/fr/xbrl/func/func.xsd*.

In interval arithmetic each reported number is converted to an interval based on it expression (reported value) and precision (*@decimals* attribute[26]) as exemplified in *Table 6*.

Table 6. Examples of intervals.

Reported number	@decimals	Precision	Interval
123 456.789	-3	in thousands (+/- 500)	(122 956.789; 123 956.789)
	0	in units (+/- 0.5)	(123 456.289; 123 457,289)
	2	to two digits after decimal point (+/-0.005)	(123 456.784; 123 456.794)
	INF	exact (+/- 0)	(123 456.789; 123 456.789)

Following that conversion, the interval arithmetic functions use basic operations (as implemented in *https://www.eurofiling.info/eu/fr/xbrl/func/interval-arithmetics.xml*) to compute the resulting intervals after applying mathematical operations. For instance in case of addition of two numbers *A* and *B*, where *A* is interval of *(A1;A2)*, *B* is interval of *(B1;B2)* the result is interval of *(A1+B1;A2+B2)*. If the interval of the reported number

---

[26]   Or *@precision* attribute which is however prohibited by the EIOPA XBRL Filing Rules published on https://eiopa.europa.eu/regulation-supervision/insurance/reporting-format.

overlaps with the computed interval the assertions is satisfied. An example in *C = A + B*, where:

- *A* is reported as 1499 with precision in units (*@decimals = 0*) hence the resulting range is (1498.5;1499.5),

- B is reported as 1502 with precision in units (*@decimals = 0*) hence the resulting range is (1501.5;1502.5),

- C is reported as = 3000 with precision in units (*@decimals = 0*) hence the resulting range is (2999.5;3000.5).

Following the basic operations, the computed tolerance interval for *A + B* is (1498.5+1501.5;1499.5+1502.5) = (3000;3002). As presented on *Figure 1* there is an overlap (marked in orange) between the interval of *C* (in blue) and interval of *A + B* (in green). As a result the assertion is satisfied.
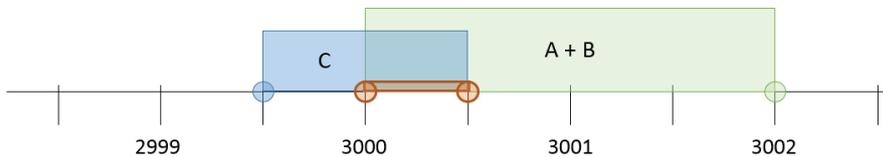
Figure 1. Overlap of intervals

If *C* was reported as 2999, the resulting interval (with precision in units) would be (2998.5;2999.5). With no overlap (see *Figure 2*) the assertion would not be satisfied and an error would be raised.
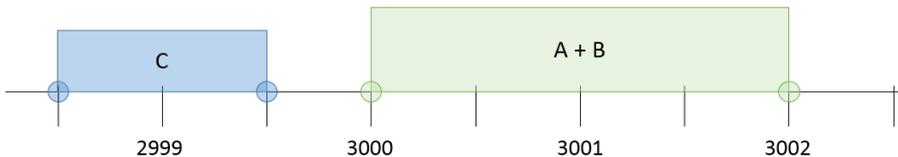
Figure 2. No overlap of intervals

Implementation of interval arithmetic defines the following functions:

- *iaf:sum*,

- *iaf:numeric-equal*,

- *iaf:numeric-less-than, iaf:numeric-less-equal-than*,

- *iaf:numeric-greater-than, iaf:numeric-greater-equal-than*,

- *iaf:numeric-add*,

- *iaf:numeric-subtract*,

- *iaf:numeric-divide*,

- *iaf:numeric-multiply*,

- *iaf:abs, iaf:min, iaf:max*,

where for example:

- *iaf:numeric-equal(arg1, arg2)*: returns *true* if two values are equal or are within the tolerance interval derived from its reported precision,

- *iaf:numeric-less-than(arg1, arg2)*: checks whether *arg1* is less than *arg2*, considering their precision.

In more complex expressions functions are nested following the order of their executions. For example *a = ((b − c) / (d * e)) + b* would be defined as *iaf:numeric-equal($a,iaf:sum((iaf:numeric-divide(iaf:numeric-subtract($b,$c),iaf:numeric-multiply($d,$e))),$b))*.

9.5.9 Deactivations

The deactivation is managed in set folder, it is stored in {module code}-ignore-val.xml file.

Each deactivated assertion is linked to a "false" precondition to prevent it being evaluated , for example:

```
<gen:link xlink:type="extended" xlink:role="http://www.xbrl.org/2003/role/link">
    <variable:precondition xlink:type="resource" xlink:label="ignore" test="false()"/>
    <link:loc xlink:type="locator" xlink:href="../val/vr-v11331_s.xml#eba_v11331_s"
    xlink:label="loc_eba_v11331_s"/>
    <gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
    xlink:from="loc_eba_v11331_s" xlink:to="ignore"/>
    <link:loc xlink:type="locator" xlink:href="../val/vr-e4435_e.xml#eba_e4435_e" xlink:label="loc_eba_e4435_e"/>
    <gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition"
    xlink:from="loc_eba_e4435_e" xlink:to="ignore"/>
</gen:link>
```

9.5.10 Version numbers

The taxonomy package provided by EBA and EIOPA will contain the version number assigned to modules and dictionaries, for example:

```
<?xml version="1.0" encoding="UTF-8"?>
<taxonomyPackage xml:lang="en" xmlns="http://xbrl.org/2016/taxonomy-package"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xbrl.org/2016/taxonomy-package
http://www.xbrl.org/2016/taxonomy-package.xsd">

    <identifier>http://www.eba.europa.eu/eu/fr/xbrl/tp/EBA_CRD_XBRL_Reporting_Frameworks_4.5.0.zip</identifier>
    <name>EBA CRD XBRL Reporting Frameworks 4.5.0</name>
    <description>EBA CRD XBRL Reporting Frameworks 4.5.0.0 - COREP, FINREP. Requires Dictionary 4.5 or later</description>
    <version>4.5.0</version>
    <publisher>European Banking Authority</publisher>
    <publisherURL>http://www.eba.europa.eu/</publisherURL>
    <publicationDate>2024-09-30</publicationDate>

    <entryPoints>
        <entryPoint>
            <name>COREP_ALM: Additional Liquidity Monitoring - COREP</name>
            <version>4.5.0</version>
            <entryPointDocument href="http://www.eba.europa.eu/eu/fr/xbrl/crr/fws/corep/4.5.0/mod/corep_alm.xsd"/>
            <languages>
                <language>en</language>
            </languages>
        </entryPoint>
        <entryPoint>
        <entryPoint>
        <entryPoint>
        <entryPoint>
        <entryPoint>
        <entryPoint>
</taxonomyPackage>
```

10 Hypercubes

It is important to remark that the XBRL hypercubes in the taxonomy are validation artefacts (essentially just indicating grey cells) and should not be used by external systems for the automatic creation of database structures. The hypercubes in the taxonomy are generated automatically by an algorithm, and do not obey to any kind of business criteria. These hypercubes might be dramatically modified with any future change to the reported information in a table, with the only consideration being the reduction of the final set of hypercubes and performing more efficiently with XBRL market tools.

11 Appendix 1: additional arcroles used

The following arcroles from the Eurofiling model.xsd are used to improve the validation of reports and to provide additional information on the constructs defined in the taxonomy and their relationships.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/complete-breakdown
Id	complete-breakdown
Definition	Member is equal to the aggregation of its breakdown
Used on	Calculation linkbase
Rationale	To enable software to correctly evaluate the relationship.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/partial-breakdown
Id	partial-breakdown
Definition	Member is equal or greater than the aggregation of its breakdown
Used on	Calculation linkbase
Rationale	To enable software to correctly evaluate the relationship.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/superset-breakdown
Id	superset-breakdown
Definition	Member is equal or less than the aggregation of its breakdown
Used on	Calculation linkbase
Rationale	To enable software to correctly evaluate the relationship.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/group-table
Id	group-table

Definition	Table group is the parent of other table groups and/or other tables
Used on	gen:arc
Rationale	To enable applications to correctly render the data in a report.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/applies-to-table
Id	applies-to-table
Definition	The list of assertions associated to an assertion set are applied to the table / tables pointed by this arc
Used on	gen:arc
Rationale	To assist software in determining which validation rules to run, given the data present in the report as indicated by the filing indicators.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/previous-version
Id	previous-version
Definition	This arc connects a specific version of an enumerated metric to the previous version of the same metric, having different allowed values
Used on	Definition linkbase
Rationale	To ensure a user using data from multiple reports based on different taxonomies, that these enumerated metrics are related. The user must determine whether this data can be combined for the intended purpose.

arcRoleURI	http://www.eurofiling.info/xbrl/arcrole/initial-version
Id	initial-version
Definition	This arc connects a specific version of an enumerated metric to the initial version of the same metric, having different allowed values
Used on	Definition linkbase
Rationale	To ensure a user using data from multiple reports based on different taxonomies, that these enumerated metrics are related. The user must determine whether this data can be combined for the intended purpose.

12 Appendix 2: additional roletypes used

All roletypes are added to group similar relationships and identify their use.

RoleType	http://www.eurofiling.info/xbrl/role/filing-indicator-code
Id	filing-indicator-code
Definition	The code to be used in a filing indicator to indicate that a "filing unit" (usually a template) has (or has not) been reported. Used as a label applied to a table:table node.
Used on	label:label, link:label

RoleType	http://www.eurofiling.info/xbrl/role/rc-code
Id	rc-code
Definition	Optional numeric code applied to the components of columns, rows and z-axes in tables, as represented in business templates
Used on	label:label, link:label

RoleType	http://www.eurofiling.info/eu/fr/xbrl/ext/BlockDefaultUseOfMetrics
Id	BlockDefaultUseOfMetrics
Definition	Prevents default use of metrics (i.e. when not explicitly allowed)
Used on	link:definitionLink

RoleType	http://www.eurofiling.info/eu/fr/xbrl/ext/-BlockDefaultUseOfMetricsScenario
Id	BlockDefaultUseOfMetricsScenario
Definition	Prevents default use of metrics in the scenario element (i.e. when not explicitly allowed)
Used on	link:definitionLink

RoleType	http://www.eurofiling.info/eu/fr/xbrl/ext/-BlockDefaultUseOfMetricsSegment
Id	BlockDefaultUseOfMetricsSegment
Definition	Prevents default use of metrics in the segment element (i.e. when not explicitly allowed)
Used on	link:definitionLink

RoleType	http://www.eurofiling.info/eu/fr/xbrl/ext/-EnumeratedMetricPreviousVersion
Id	EnumeratedMetricPreviousVersion
Definition	Linking an enumeration to the previous version of the same enumerated metric
Used on	Definition linkbase

RoleType	http://www.eurofiling.info/eu/fr/xbrl/ext/-EnumeratedMetricInitialVersion
Id	EnumeratedMetricInitialVersion
Definition	Linking an enumeration to the initial version of the same enumerated metric
Used on	Definition linkbase

13 Appendix 3: file structure

Architecture file structure

{owner location}	{owner namespace}	{owner prefix}
http://www.eurofiling.info/eu/fr/xbrl	http://www.eurofiling.info/xbrl/	eu
http://www.eba.europa.eu/eu/fr/xbrl/crr	http://www.eba.europa.eu/xbrl/crr	eba
http://eiopa.europa.eu/eu/xbrl	http://eiopa.europa.eu/xbrl	eiopa

{owner location}

dict

met

met.xsd,
met-lab-en.xml

{owner prefix}_met
{owner namespace}/dict/met

Version number

met.xsd (enumerated)
met-lab-en.xml (enumerated)
hier.xsd, hier-lab-en.xml, hier-def.xml

{owner prefix}_h
{owner namespace}/dict/met/version_number/hier

{owner prefix}_met_version_number
{owner namespace}/dict//met/version_number

dim

version number

dim.xsd,
dim-lab-en.xml,
dim-def.xml

{owner prefix}_dim
{owner namespace}/dict/dim/version_number

dom

exp.xsd,
exp-lab-en.xml

{owner prefix}_exp
{owner namespace}/dict/exp

{owner prefix}_typ, {owner namespace}/dict/typ

typ.xsd,
typ-lab-en.xml

{owner prefix}_{DC}
{owner namespace}/dict/dom/{DC}

{dc}
(domain code)

mem.xsd,
mem-lab-en.xml

version number

hier.xsd
hier-lab-en.xml
hier-def.xml

{owner prefix}_{DC}_h
{owner namespace}/dict/dom/{DC}/{version_number}/hier

fws

fws.xsd,
fwr-lab-en.xml

{owner prefix}_fws
{owner namespace}/fws

{framework}
(framework name)

Version number

tab

tab.xsd,
tab-lab-en.xml

{owner prefix}_tab
{owner namespace}/fws/{fws}/{pub-date}_tab

{table}

{table}.xsd,
{table}-lab-en.xml,
{table}-lab-codes.xml,
{table}-def.xml,
{table}-rend.xml

{owner prefix}_tab_{table}
{{owner namespace}/fws/{fws}/version_nur

mod

{module}.xsd
{module}-lab-en.xml
{module}-pre.xml
{module}.son

{owner prefix}_mod_{module}
{{owner namespace}/fws/{fws}/version_number /mo

val

val-{assertion code}.xml
val-{assertion code}-lab-en.xml
val-{assertion code}-err-en.xml
params.xml,
params-lab-en.xml, find-params.xml,
{module}-val.xsd, {module}-find-
prec.xml. {module}-val-tabs.xml

{owner prefix}_val_{module}
{{owner namespace}/fws/{fws}/version_numb

set

{module}-val-severity.xml
{module}-ignore-val.xml

ext

model.xsd

model
http://www.eurofiling.info/xbrl/ext/model

func

functions.xsd,
interval-arithmetics.xml
params.xml

func
http://www.eurofiling.info/xbrl/func

S