

RECEiX HANDBOOK



Version update 2025-01-13 patch 2025-07-07. Past versions 2018-12-15, 2018-03-18

Copyright (Licensors): Javier Mora, Ignacio Boixo at boixo@eurofiling.info

MANDATORY CITATION. License Creative Commons Attribution 4.0 International



ABSTRACT:

The Really Easy Converter from Excel to iXBRL (*RECEiX*) is an Excel proof-of-concept open source tool with an input of Excel spreadsheets and an output of (a) the corresponding extended XBRL taxonomy and (b) the corresponding iXBRL and XBRL documents.

RECEiX fully follows the XHTML 1.1 and XBRL specifications, as inline XBRL, Dimensions and Taxonomy package, as well as Anchoring draft.

The iXBRL document can be displayed using whatever Internet browser (Microsoft Explorer, Mozilla Firefox, Google Chrome...) with the same layout that the Excel spreadsheets.

The iXBRL document can be also processed by whatever XBRL processor, as Arelle open source suite, Openfiling iXBRL extractor, or any other XBRL tool.

OBJECTIVES:

The first objective is didactic, for software developers when creating iXBRL documents and extended XBRL taxonomies from scratch.

The second objective is providing empirical evidence, for reporting entities and regulators, about iXBRL generation and consumption.

The third objective is contributing to the European Securities and Markets Authority (ESMA) efforts, initially on the European Single Electronic Format (ESEF), according to the European Parliament decisions about transparency in financial markets.

REQUISITES:

Any Microsoft Excel version with VBA enabled on Windows. Standalone use: Internet is not required.

AUTHORS:

Ignacio Boixo as Excel VBA developer. Javier Mora as XBRL expert. [Openfiling Team for ESEF](#).

REFERENCES:

[XBRL standards](#), [ESMA ESEF website](#), [Eurofiling ESEF premier](#) and www.openfiling.info

DISCLAIMER:

Unless otherwise separately undertaken by the Licensors, to the extent possible, the Licensors offer RECEiX as-is and as-available, and make no representations or warranties of any kind concerning RECEiX, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. To the extent possible, in no event will the Licensors be liable on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this license or use of the RECEiX, even if the Licensors have been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability by law is not allowed in full or in part, this limitation may not apply. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Content

Process description:	3
iXBRL essentials	4
HTML and XBRL	4
Inline XBRL.....	5
From Excel to iXBRL instance document	5
From Excel to XBRL extension taxonomy	6
iXBRL instance document.....	7
Generating iXBRL stuff	8
Columns in <i>data</i> spreadsheet:	8
Skip.....	8
Qname	8
Fact.....	8
PeriodInit	9
PeriodEnd	9
UnitRef (Fixed, Numeric, Date, Enumeration)	9
Decimals.....	12
Scale.....	12
Dimension DimensionMember	12
Parameters	13
Mandatory parameters.....	13
Optional parameters.....	14
Parameters for taxonomy extension only.....	15
Additional optional parameters, for expert use only.....	16
Footnotes spreadsheets (optional).	17
Errors sheet.....	18
Exported Path and File name	18
Favicon (optional).....	18
Validator (optional).....	18
Taxonomy extension (optional).....	19
Taxonomy extension directories and files	19
Columns in the dictionary spreadsheet.....	20
Calculation and Presentation: Parent.....	22
Dimension: Parent.....	23
Presentation, Indent, NestLevel optional columns:.....	24

Process description:

Simply click in the button “Create iXBRL document and extension taxonomy” and wait while processing. Seconds later, a pop-up message will inform you about the creation of both the iXBRL document and the taxonomy extension files, or an error message instead. The processing time varies from few seconds for a simple test to several minutes for 50,000+ cells.

RECEiX is a simple Excel workbook with a number of “entry” spreadsheets populated of financial information and up two spreadsheets (“data” and “dictionary”) with XBRL declarative information.

The “entry” spreadsheets are designed by the accountant (with the optional help of a designer) for each specific use case, including financial statements, notes, and so on. RECEiX transforms these “entry” spreadsheets into a single .html file, displayable in any [web browser](#). The transformation concatenates all the visible “entry” spreadsheets of the Excel book, separated by horizontal lines. Cells contents, links, sizes, merges, borders, colours and other decoration attributes are maintained. Images in the “entry” spreadsheets are generated as embedded base64 stuff in the .html file. Spreadsheets process is by order in Excel, from top-left, columns and then rows.

The “data” spreadsheet is prepared by the accountant and/or the IT expert, by linking each financial data appearing in the “entry” spreadsheets with the corresponding tag (base or extension taxonomy item) plus the specific attributes of use, as period, accurate decimals, currency, dimensions and so on. Note that several financial data could be identically tagged but with different attributes (e.g. period this year / period past year). All these “data” spreadsheet stuff is included as iXBRL metadata embedding each financial data included in the .html file. Then the .html file is now iXBRL compliant as iXBRL instance document.

The “dictionary” spreadsheet is tagged (choosing the most appropriate element in a dictionary of accounting terms that best represents each financial data) by the accountant (with the optional help of an IT expert). The tagging is picking up the IFRS/GAAP accounting concepts needed (base taxonomy items) and, for the tags not found, create the required additional accounting concepts used proprietarily by the issuer (extension taxonomy items). RECEiX creates the corresponding *Extension Taxonomy* files as derivate from the “dictionary” spreadsheet. This step is optional when taxonomy extension is not required.

RECEiX checks the coherence of the input data, generating the files only if no errors detected.

In the original RECEiX example, it is used a reduced subset the IFRS 2016 base taxonomy extended for a subset of the ERICA financial statement format¹ populated with data from an anonymized financial statement, for explanatory purposes only.

RECEiX is GENERIC and independent of whatever particular XBRL taxonomy or accounting standard. The specifications used are inline XBRL, XBRL, Dimensions and Taxonomy package, as well as Anchoring draft. Some features are not yet implemented, as such features not used in ESEF (i.e. tuples) or some infrequent errors detection or some infrequent Item Types.

RECEiX is powered by a macro VBA program of about 3000 lines, of which a number are comments for self-documentation. Do not hesitate contact the authors for any questions.

¹ The reduced format has been defined by the ERICA (European Records of IFRS Consolidated Accounts) Working Group, whose members are the Central Balance Sheet Data Offices of Austria, Belgium, France, Germany, Greece, Italy, Portugal and Spain and ECB. IFRS Foundation is an observer of the ERICA WG. See more at European Committee of Central Balance-Sheet Data Offices (ECCBSO) www.eccbso.org

iXBRL essentials

The underlying challenge is the difference behaviour between humans and computers when reading a financial statement.

The humans capture the context of each value, as Entity, Period, Currency, Units and so on. Customarily, the context information is distributed in more or less predefined areas.

The front page usually contains the identification of the Entity, Currency, Period and Scale (figures in units, thousands or millions). Each statement is placed in an independent page, by rows, with the tag at the left, notes in the middle and figures at right in two columns: current year and past year.

The computers, typically an Extract/Transform/Load (ETL) process, are simply unable of contextualizing the figures in the financial statements. This is the main reason for XBRL: provide the context of all and every figure in a financial statement.

HTML and XBRL

See below a very simplified example in Excel as translated into HTML (Hyper Text Mark-up Language) for human presentation in a web browser and XBRL (eXtended Business Reporting Language) for ETL processing:

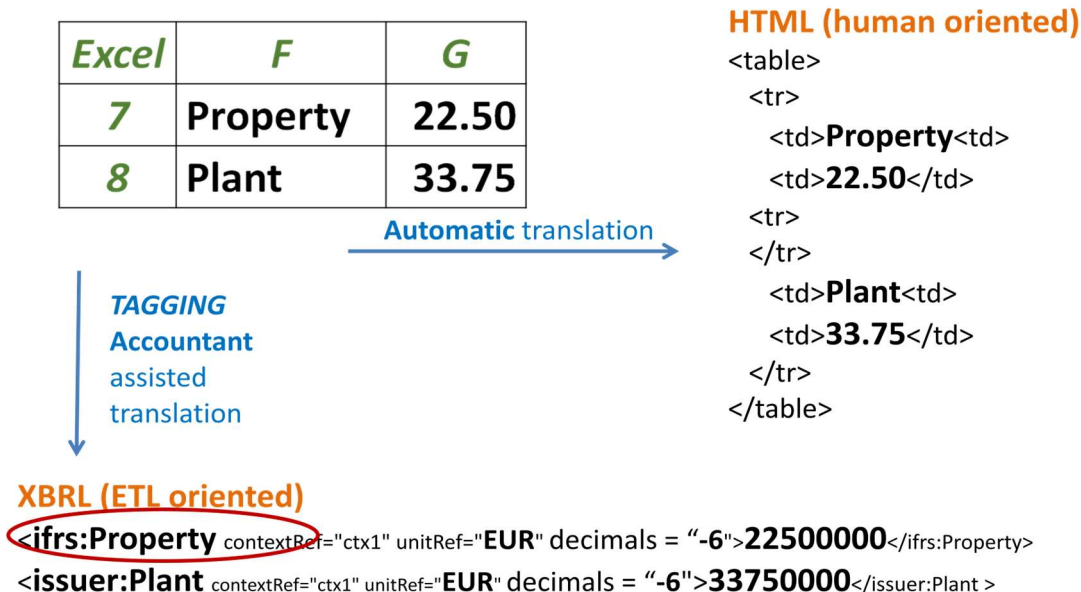


Figure 1: From Excel to HTML and XBRL

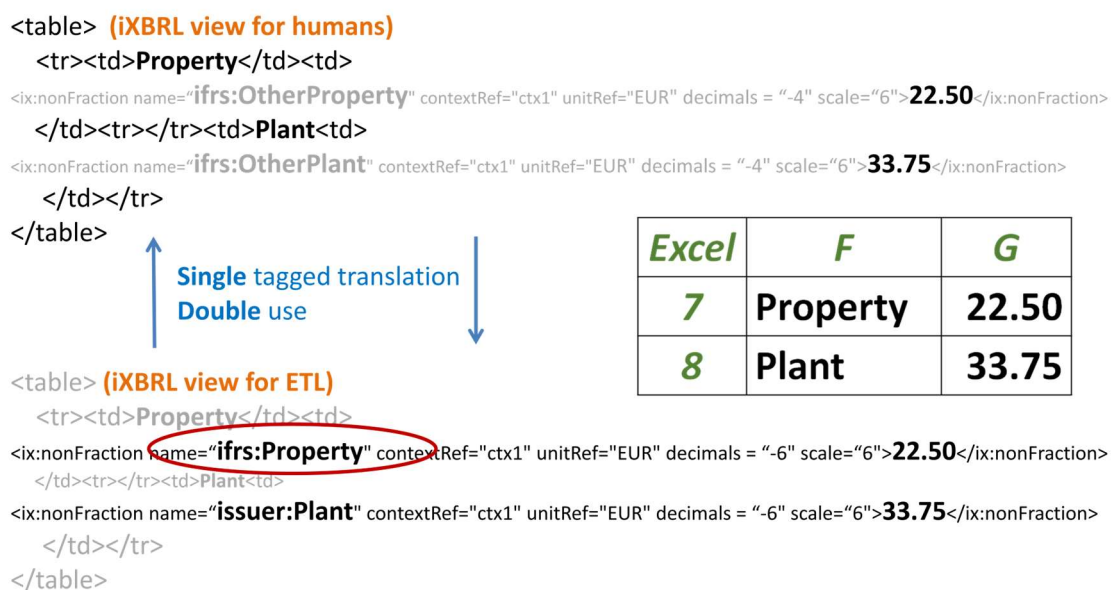
The translation from Excel to HTML is trivial: simply taking care of sheets “<table>”, rows “<tr>”, columns “<td>” and some decoration (size, colour and so on, not represented in the example).

The translation from Excel to XBRL is more complex, as each figure requires an association (metadata) with the accounting definition (tag) and contextual information. Moreover, during the translation to XBRL the presentation references for tables, rows and columns are lost.

Inline XBRL

iXBRL is a combination of HTML and XBRL based in a common characteristic shared by all the XML-derivate languages: If a program cannot interpret a particular structure, simply ignores it. Then, each figure is nested by two structures: an inner XBRL structure for ETL process (ignore by browsers) and an outer HTML structure for browser presentation (ignored in turn by ETL process).

See the next diagram as example:



Depending on the program, the same iXBRL instance document is interpreted as HTML for humans (ignoring XBRL) or XBRL for ETL process (ignoring HTML).

From Excel to iXBRL instance document

In a nutshell, this process translates the Excel financial statement directly to HTML (syntactical translation), but requires additional information for the XBRL structures.

This additional information is placed in a special sheet, the sheet "Data", with a row with the contextual meta-information for each figure (fact). The address of the Excel cell links with their definition in the sheet "Data". QName is the unique XML name of each accounting definition.

See more explanations in the following chapters.

	C	E	F	H	I
10	I. ASSETS, NON-CURRENT, TOTAL	1.66, 31.56	IFRS	95980120	93984712
11					
12	1. Property, plant and equipment	1.54.a, 16.73.d	IFRS	63834384	61788590
13	1.1. Land and buildings	16.37.b	IFRS	1798553	1431565
14	1.2. Plant and equipment	16.73.e	CBSO-RF	54673103	54701109
15	1.3. Remaining property, plant and equip	16.37	CBSO-RF	636055	468919
16	1.4. Construction in progress and	16.73.e	IFRS	6726673	5186997

Data sheet
Qname, fact and context of each cell.
iXBRL file is a direct syntactical translation.

=Assets!H10

Qname	Fact	PeriodEnd	UnitRef
ifrs-full:NoncurrentAssets	95980120	2016-12-31	iso4217:EUR
ifrs-full:NoncurrentAssets	93984712	2015-12-31	iso4217:EUR
ifrs-full:PropertyPlantAndEquipment	63834384	2016-12-31	iso4217:EUR
ifrs-full:PropertyPlantAndEquipment	61788590	2015-12-31	iso4217:EUR
ifrs-full:LandAndBuildings	1798553	2016-12-31	iso4217:EUR
ifrs-full:LandAndBuildings	1431565	2015-12-31	iso4217:EUR
cbso-rf:PlantAndEquipment	54673103	2016-12-31	iso4217:EUR
cbso-rf:PlantAndEquipment	54701109	2015-12-31	iso4217:EUR
cbso-rf:RemainingPropertyPlantAndEquip	636055	2016-12-31	iso4217:EUR
cbso-rf:RemainingPropertyPlantAndEquip	468919	2015-12-31	iso4217:EUR

From Excel to XBRL extension taxonomy

In a nutshell, this process creates an XBRL extension taxonomy for the tags used in the financial statement.

The IFRS or a particular GAAP taxonomy is used as the XBRL base taxonomy, describing all the characteristics of each accounting definition in XBRL terms, as Instant/Period, Credit/Debit, Monetary/Text and so on.

But not all the required tags are described as in the base taxonomy, basically because each user has the freedom to create specific tags for its specific business characteristics.

A “dictionary” sheet is required describing all the Qnames used in the “Data” sheet, being the “new” Qnames for the extension taxonomy and their relations with the “old” Qnames of the base taxonomy, as well as the relations among them.

In the example, see this structure of Qnames is mixing the ifrs-full base taxonomy with the cbso-rf extension taxonomy:

```
ifrs-full:PropertyPlantAndEquipment
ifrs-full:LandAndBuildings
cbso-rf:PlantAndEquipment
cbso-rf:RemainingPropertyPlantAndEquipment
ifrs-full:ConstructionInProgress
```

Note the tags cbso-rf:PlantAndEquipment and cbso-rf:RemainingPropertyPlantAndEquipment are specific for this financial statement, lacking of exact equivalence in ifrs-full taxonomy. However, the sum of the facts values for the four children must match with the fact value of the parent ifrs-full:PropertyPlantAndEquipment.

Thanks to such conventions, the information required for the generation of the XBRL extension taxonomy is described in the sheet “Dictionary”. See more explanations in the following chapters.

	C	E	F	H	I
10	I. ASSETS, NON-CURRENT, TOTAL	1.66, 31.56	IFRS	95980120	93984712
11					
12	1. Property, plant and equipment	1.54.a, 16.73.d	IFRS	63834384	61788590
13	1.1. Land and buildings	16.37.b	IFRS	1798553	1431565
14	1.2. Plant and equipment	16.73.e	CBSO-RF	54673103	54701109
15	1.3. Remaining property, plant and equip	16.37	CBSO-RF	636055	468919
16	1.4. Construction in progress and	16.73.e	IFRS	6726673	5186997

	Qname	Period	Balance	ItemType	Weight	Parent
34	ifrs-full:LandAndBuildings	Instant	Debit	Monetary	1	ifrs-full:PropertyPlantAndEquipment
35	cbso-rf:PlantAndEquipment	Instant	Debit	Monetary	1	ifrs-full:PropertyPlantAndEquipment
36	cbso-rf:RemainingPropertyPlantAndEquipment	Instant	Debit	Monetary	1	ifrs-full:PropertyPlantAndEquipment
37	ifrs-full:ConstructionInProgress	Instant	Debit	Monetary	1	ifrs-full:PropertyPlantAndEquipment
38	ifrs-full:PropertyPlantAndEquipment	Instant	Debit	Monetary	1	ifrs-full:NoncurrentAssets
54	ifrs-full:NoncurrentAssets	Instant	Debit	Monetary	1	ifrs-full:Assets

Dictionary sheet

Define the specific accounting plan for this report, aka *taxonomy extension*

Dictionary and Data sheets are indexed by Qnames

Dictionary generates Calculation, Label (multilingual), Anchoring....

Figure 4: From Excel to XBRL extension taxonomy

iXBRL instance document

In this step, RECEiX creates an iXBRL instance document, which is an *.html* file with iXBRL metadata embedded in each financial data (enabling the Extracting / Transform / Load process of such financial data). This *.html* file can be displayed using whatever web browser (Microsoft Explorer, Mozilla Firefox, Google Chrome...). Each *entry* spreadsheet cell referenced in the *data spreadsheet* is detected as XBRL fact, hence generating the corresponding iXBRL properties (metadata) to be stored in the *.html* file, but the browser does NOT display such XBRL metadata.

The starting point is a number of *entry* spreadsheets in the RECEiX Excel book, containing the original financial data (facts). An *entry* spreadsheet is whatever Excel spreadsheet in the book except the *data*, *dictionary* and *error* spreadsheets. RECEiX generates an *.html* file (iXBRL instance document) with the same layout and content that the *entry* spreadsheets, but ignoring *data*, *dictionary* and *error* spreadsheets.

The spreadsheets that are not visible in the Excel book, as well as the rows and columns that are not visible in its Excel spreadsheet are completely ignored. Simply made the Excel spreadsheet not visible, or made rows or columns not visible, and all this stuff will be absolutely ignored (also for iXBRL), as inexistent stuff.

A mixed option is including the word “*hidden*” in any position of an *entry* spreadsheet name. Such *hidden entry* spreadsheet will still generate all the iXBRL stuff (facts and footnotes as well) in the iXBRL instance document, but in an *.html* hidden (not-visible) mode, being such iXBRL data not displayable by any web browser. From the iXBRL perspective, the iXBRL processors only get the data embed into iXBRL metadata, irrespective if such data is visible or not for the web browser of any other *.html* presentation characteristic.

The *.html* file will maintain the layout and decoration of each *entry* spreadsheet cell, irrespective if detected as XBRL fact or not. The best practical way for observing XBRL stuff in the *.html* file is selecting the option viewer in the *data spreadsheet* parameter */**Options*.

The attributes of size, border, colour, alignment and so on of each cell are copied from Excel as far as RECEiX can. The images in Excel are copied as embedded `` on base64 (with size limitation of 65535). Hyperlinks are opened in the same window (the best practice is click in the mouse middle button for opening in a new window). RECEiX only generates in *.html* the rectangle of rows and columns of each *entry* spreadsheet containing the actually used cells (even with blanks). RECEiX optimizes the resulting *.html* file by using `colspan` and `rowspan` for grouping empty cells and selecting the most used option of each attribute as CSS default.

The *.html* file containing the iXBRL instance document is generated with *.html* file type extension. In case of taxonomy extension, see about *taxonomy package* directory.

RECEiX can create also (or instead) an equivalent XBRL instance document, only with the XBRL information, without *.html* presentation stuff. The file containing the XBRL instance document will be generated but with the file type extension *.xbrl* as the XBRL carbon copy of the iXBRL instance document, but now using the XBRL 2.1 syntax instead of the iXBRL syntax.

Generating iXBRL stuff

Each *entry* spreadsheet cell with iXBRL fact (financial data) is referenced by a row in the *data* spreadsheet, being such row the iXBRL definition of cell. Only the *entry* spreadsheet cells referenced in the *data* spreadsheet are used for XBRL purposes: any other cell is ignored in XBRL consequently.

Each row (not starting in “/*”) in “*data*” spreadsheet MUST contain, in the column “Fact”, an Excel reference to the corresponding iXBRL cell in the *entry* spreadsheets (E.g.: “=Assets!\$H\$10”). In this way, RECEiX is able to determine the iXBRL properties applicable to each original fact and produce a fact in the iXBRL document.

Each row in the “*data*” spreadsheet contains all the information required to generate a fact in the XBRL instance document. The taxonomy information is not used in this step (except for some crossed checks). Yes, some information in *data spreadsheet* may be redundant, but the resulting VBA code is more didactic and easier to reuse.

Simply open the exported iXBRL document (*xhtml* file) using whatever Internet browser (Microsoft Explorer, Mozilla Firefox, Google Chrome...). The data is shown it also can be re-used with the standard “copy and paste” functions.

Columns in *data* spreadsheet:

The meaning of the *data* spreadsheet columns roughly follows [Inline XBRL 1.1](#) specification.

The columns in the *data* spreadsheet are placed in fixed positions (wired in the VBA code).

This part of RECEiX is an evolution of RECEX (Really Easy Converter Excel to XBRL), dated 2011. See additional rationale, examples and more details about the use of the *data* spreadsheet at www.openfiling.info/receX/ as well as instructions and UML description for XBRL instance documents at www.openfiling.info/wp-content/uploads/data/RECEX-instructions.pdf

Skip

Optional. If “/*” this row is not processed for Fact. Used for comments, parameters, or simply for skip this row.

Qname

Mandatory. If the *dictionary* spreadsheet exists, this Qname must be on it.

Fact

Mandatory. Must be an Excel reference to a cell in an *entry* spreadsheet containing the value of the XBRL fact. e.g. =Assets!\$H\$10

This Excel reference is used to locate the *entry* spreadsheet cells to be wrapped with the corresponding iXBRL metadata stuff. Hence, the fact value generated in *.html* is embedding into iXBRL metadata. If the referenced *entry* spreadsheet cell is empty, the iXBRL metadata is not created.

Numeric facts:

Negative figures would be preceded and/or followed by one (or a string) of the symbol “-” as in “-10”, or placed between brackets as in “(-10)”. Note “(-10)” is positive.

The use of thousand separators (as “.”) is optional

Leading (at left) or tailing (at right) symbols are preserved as *html* decoration, but not used for computing numeric values, as “#####10.00#” or “ -10 \$” or other combinations. Rule of thumb: The only symbols allowed between a digit and the following digit are “,” and “.” (*comma* and *dot*) as thousands and decimal separators.

Excel uses dot or comma as decimal or thousands separator respectively, depending on the country configuration. See more ahead in *Optional Parameters / **Options*

Exponential notation as “6.464E+21” is not allowed. Note that Excel sometimes uses the exponential notation if the cell has not enough space for the figure.

Use of indent and spaces for alignment:

Excel indentation is allowed and translated as *html padding-left (number of indents)*

Leading (at left) or tailing (at right) spaces are ignored unless used for cell alignment.

Left alignment preserves leading spaces (at left) while right alignment preserves tailing spaces (at right), replacing spaces by symbols *html * (former symbol * *);

Indent and spaces for alignment are compatible in the same cell, and generally (not always) are functionally equivalent. The recommended best practice is indentation.

PeriodInit

Optional. If used, put here the *startDate*. The assumed period type in this case is *duration*. If left empty, then the assumed period type is *instant*.

Default date format is YYYY-DD-MM.

Note: If the *PeriodInit* or the *PeriodEnd* value is copied from other cell in the sheet data (the the *PeriodInit* or the *PeriodEnd* cell contains the address of a fact cell in sheet data, as e.g. = $\$C\25), and the UnitRef of the addressed cell (e.g. row 25) is a date format (e.g.: datemonthdayyear), then such date formats is used.

This arrangement is useful when a Fact contains the value of the *PeriodInit* or the *PeriodEnd* and, in general, for better visualizing the date to be used in periods.

PeriodEnd

Mandatory. Used as *endDate* if the assumed period type is *duration* (i.e. *PeriodInit* is not empty). Otherwise used as *instant* date (i.e. *PeriodInit* empty).

Default date format is YYYY-DD-MM.

(See note on *PeriodInit*, also applicable here).

UnitRef (Fixed, Numeric, Date, Enumeration)

Fixed Types [true|false|zero|empty]

Numeric Types [Monetary|Integer|Pure|Shares|PerShare|Percent]

Date Types [dateyearmonthday|datemonthdayyear|datedaymonthyear]

Enumeration [enum:='sheet'! $\$C\$n:\$D\m]

Otherwise let in blank.

Fixed Types [true|false|zero|empty]

See XBRL Specification [Transformation Registry 5](#)

Functions: ixt-fixed-true.xml, ixt-fixed-false.xml, ixt-fixed-zero.xml, ixt-fixed-empty.xml

The resulting fixed Type is included into the iXBRL schema as follows:

```
<ix:nonNumeric name="tax:Concept" contextRef="_ctx" id="factID"
  format="ixt:fixed-true" escape="false">Loren ipsum</ix:nonNumeric>
```

The resulting XBRL schema is as follows:

```
<tax:Concept" contextRef="_ctx" id="factID">true</tax:Concept>
```

Numeric Types [Monetary|Integer|Pure|Shares|PerShare|Percent]

If the UnitRef value is explicit (following an XBRL notation as xbrli:pure or iso4217:EUR or xbrli:shares), then RECEiX uses the value for generating appropriate *xbrli:unit* expression in the output file. For example, the explicit UnitRef *xbrli:pure* produces:

```
<xbrli:unit id="XBRLI_PURE"><xbrli:measure>xbrli:pure</xbrli:measure></xbrli:unit>
```

The separator “/” denotes an explicit declaration of divide UnitRefs, by concatenating the numerator UnitRef and the denominator UnitRef. For example: *iso4217:EUR/xbrli:shares* which produces:

```
<xbrli:unit id="ISO4217_EUR_XBRLI_SHARES">
  <xbrli:divide>
    <xbrli:unitNumerator>
      <xbrli:measure>iso4217:EUR</xbrli:measure>
    </xbrli:unitNumerator>
    <xbrli:unitDenominator>
      <xbrli:measure>xbrli:shares</xbrli:measure>
    </xbrli:unitDenominator>
  </xbrli:divide>
</xbrli:unit>
```

For expert use only: Other option is placing here a UnitRef “id” (without any “:” separator) and placing the corresponding full UnitRef declaration for such UnitRef “id” in a row using the */**Resources* parameter.

Date formats [dateyearmonthday|datemonthdayyear|datedaymonthyear]

See XBRL Specification [Transformation Registry 5](#)

Date format is YYYY-MM-DD or MM-DD-YYYY or DD-MM-YYYY respectively.

Default date format is dateyearmonthday

Accepted separators are any combination of [space|/|-|_|.|*|st|nd|th]

If year is expressed as YY, then year is expanded to 20YY

If month is expressed as M, then month is expanded to 0M

If day is expressed as D, then day is expanded to 0D

E.g.: dateyearmonthday **18/7/1** is interpreted as **2018-07-01**

If the month is expressed as its English name (in full or only with three initial letters), then the date type is replaced by [dateyearmonthdayen | datemonthdayyearen | datedaymonthyearen] respectively.

E.g.: datemonthdayyear **Jul 1st, 18** is interpreted as datemonthdayyearen **2018-07-01**

Enumeration [enum:='sheet'!\$C\$n:\$D\$m]

See [Enumerations in XBRL](#) guidance

See XBRL Specification [Extensible Enumerations 2.0](#)

The basic concept here is using the Data Validation functionality of Excel to select a value to display in a “entry” sheet among a list of fixed values. The values to display have been previously customized in a column of an auxiliary hidden Excel sheet for enumerations. Each value to display will have predefined the enumeration value in the column A of the same row. With this arrangement, this program will obtain the enumeration value of each selected display value. In the following example, the value to display is “Belgium” in row 21, column B. The column A in the same row 21 has the corresponding enumeration value “https://xbrl.org/2024/iso3166#BE”:

	A	B
19	https://xbrl.org/2024/iso3166#BB	Barbados
20	https://xbrl.org/2024/iso3166#BD	Bangladesh
21	https://xbrl.org/2024/iso3166#BE	Belgium
22	https://xbrl.org/2024/iso3166#BF	Burkina Faso
23	https://xbrl.org/2024/iso3166#BG	Bulgaria

If this example, the “entry” sheet would have a Data List validation as follows:

The screenshot shows an Excel spreadsheet with a dropdown menu for 'Country' in cell B28. The dropdown list includes: Åland Islands, Azerbaijan, Bosnia and Herzegovina, Barbados, Bangladesh, Belgium (selected), Burkina Faso, Bulgaria, Bahrain, Burundi, Benin, and Saint Barthélemy. To the right, the Data Validation dialog box is open, showing the 'Settings' tab. The 'Validation criteria' section is set to 'List'. The 'Data' section is set to 'between'. The 'Source' field contains the formula '=enum-CountryDomain!\$B1:\$B256'. The 'Ignore blank' and 'In-cell dropdown' checkboxes are checked. The 'Apply these changes to all other cells with the same settings' checkbox is unchecked. The dialog box has 'Clear All', 'OK', and 'Cancel' buttons.

In each run of this program, the UnitRef column of sheet *Data* will be automatically updated with the corresponding Data List validations for each “entry” sheet.

Therefore, please do not create by hand the Enumeration [enum:='sheet'!\$C\$n:\$D\$m] texts in the UnitRef column of the sheet *Data*.

However, this update will not run if the “entry” sheets are protected, because programmatically reading Data List is blocked in this case, hence the UnitRef column of sheet *Data* will not be updated. The recommendation here is to run the program at least once before protecting the “entry” sheets for the initial update.

Enumeration [enum:namespace#value]

See [Enumerations in XBRL](#) guidance

See XBRL Specification [Extensible Enumerations 2.0](#)

The text in UnitRef contains the explicit enumeration string to be used, irrespective of the value of the cell in the “entry” sheet.

Example on UnitRef: “enum:https://xbrl.org/2024/iso3166#BE”

It is responsibility of the program update the enumeration value in the column UnitRef sheet *Data* if required. To avoid confusions, do not use Data List on the cell in the “entry” sheet.

Decimals

Mandatory for ItemTypes [Monetary|Integer|Pure|Shares|PerShare|Percent]. Otherwise let empty.

Scale

Mandatory for ItemTypes [Monetary|Integer|Pure|Shares|PerShare|Percent]. Otherwise let empty.

The final value in units is the result of the current value multiplied by raising ten to the power of the actual value of the scale² attribute.

Dimension DimensionMember

Optional

Pair of columns, containing respectively the actual values of Dimension and DimensionMember applicable to this fact, as in this example:

cbso:ThisAxis *cbso:ThisMember*

Use all the necessary consecutively pair of columns for several dimensions, e.g.:

cbso:FisrtAxis *cbso:MemberA* *cbso:SecondAxis* *cbso:MemberZ*

In case of typed dimensions (not allowed in ESEF), use four consecutive columns: (1) Dimension, (2) MemberType, (3) */**Typed* literal and (4) MemberValue. See the hypothetical example of a typed dimension for LEI codes:

cbso:LEIAxis *cbso>TypeLEI* */**Typed* *5493006N13E05R25BQ54*

For expert use only: Scenario/segment

By default, the dimensions are generated inside "*<scenario>*" Contextual Information, as required by ESEF.

Using the literals */**Scenario* and */**Segment*" forces that all the dimensions (if any) at the left of */**Scenario*" are to be placed as "*<scenario>*", and all dimensions (if any) at the right of */**Segment*" are to be placed as "*<segment>*"

The command */**Scenario /**Segment*" is double and therefore must be placed in two consecutive cells. Only the first occurrence of */**Scenario /**Segment*" is processed;

Tuples:

Tuples description is to be placed here, bounded and unbounded as well.

Tuples are not implemented yet in RECEiX, basically because no use case has been detected yet. For additional information and examples about tuples, see RECEX implementation of tuples at www.openfiling.info/receX/

² In financial statements, this is commonly done to allow numbers to be scaled and presented to the reader in thousands, millions or billions. Similarly, a percentage which might be displayed as an integer will often be incorporated into the XBRL Instance as a decimal. Where the scale of the displayed value in the Inline XBRL Document is different to that used in the XBRL Instance, the element is given a scale attribute to indicate the required scaling value. For instance, if the Inline XBRL Document displays a value in units of thousands, it will be necessary to multiply the value of the element by 1,000 to determine the correct value to be included in the XBRL Instance, and the scale attribute is used to establish this.

Parameters.

The parameters' information is used for the header of the iXBRL instance document.

The parameters' information is also used when optionally generating the extension taxonomy.

Parameters may be placed in whatever rows of the *data* spreadsheet after the header

Put the type of parameter (e.g. */** Entity*) in the column **Skip** (Column A), and the corresponding value in the **Fact** (Column C), (e.g. *KGCEPHLVVKVRZYO1T647*).

Mandatory parameters.

/Entity** *KGCEPHLVVKVRZYO1T647*

Identifier of the issuer entity as e.g. *KGCEPHLVVKVRZYO1T647*

The type of Identifier is usually given by the regulator.

LEI is used for ESEF reporting. In this case, RECEiX checks that the */**Entity* value is a well-formed LEI code, according to ISO 17442.

See <https://www.gleif.org/en/about-lei/iso-17442-the-lei-code-structure#>

/Xmlns** *xmlns:ifrs-full="http://xbrl.ifrs.org/taxonomy/2016-03-31/ifrs-full"*

xmlns declaration referencing the base taxonomy and other stuff

Repeat this parameter for each *xmlns* declaration

/Href** *http://xbrl.ifrs.org/taxonomy/2016-03-31/full_ifrs/full_ifrs-cor_2016-03-31.xsd*

href declaration referencing the base taxonomy

Optional parameters.

/Options** *ESEF,iXBRL, viewer, XBRL,numdotdecimal,numcommadecimal*

Select the combination that you want, separated by commas.

iXBRL Create an iXBRL instance document. If **/**Options** is not declared, the default value is iXBRL

viewer Insert the code for embedded Inline XBRL viewer. More details at filed test on ESEF <https://www.esma.europa.eu/field-test-esef>

XBRL Create an XBRL instance document

ESEF RECEiX to validate the iXBRL document according to the rules of the ESEF Reporting Manual, as published at <https://www.esma.europa.eu/document/esef-reporting-manual>

numdotdecimal instructs Excel to use `Application.DecimalSeparator="."` and `Application.ThousandsSeparator=","`

numcommadecimal instructs Excel to use `Application.DecimalSeparator=","` and `Application.ThousandsSeparator="."`

If neither **numdotdecimal** nor **numcommadecimal** are declared, the Excel's `Application.DecimalSeparator` and `Application.ThousandsSeparator` will be unchanged. If both are declared, an error will be raised.

RECEiX only accepts the pair of values `","` `."` or `."` `","` in `Application.DecimalSeparator` and `Application.ThousandsSeparator` respectively.

RECEiX will generate iXBRL with **numdotdecimal** attribute in numeric facts when Excel has `Application.DecimalSeparator="."` and `Application.ThousandsSeparator=","`

RECEiX will generate iXBRL with **numcommadecimal** attribute in numeric facts when Excel has `Application.DecimalSeparator=","` and `Application.ThousandsSeparator="."`

/ClearArea** *Index!\$F\$18:\$I\$27*

Declaration of an Excel range forced to be ignored when generating iXBRL documents.

The Excel range is assumed to be empty, irrespective the actual Excel content it may have. Useful for skipping action buttons definitions or other stuff having a meaning in Excel but having no sense in the *html* iXBRL instance document.

For declaring several Excel ranges, repeat the parameter **/**ClearArea** and/or concatenate the ranges with the separator comma `","` without spaces between them, as in `Index!F18:I20,Index!F20:I27`

/Title** *Title of the Report in the Navigator tab*

Title of the iXBRL file, to be show in the navigator tab. Default is *"RECEiX Report"*



/ScaleSheet** 125%

Percentage between 10% and 999% to scale the size (font, width, height and so on) of the iXBRL from Excel to xhtml. Default is 100%. Used to increase or decrease the size on the xhtml file in the Navigator

/IgnoreSheet** *Instructions,Help*

Sheets to be explicitly ignored when exporting the iXBRL or XBRL file. It is equivalent to hide these sheets. Default sheets are *Data* , *Dictionary* and sheet for errors. Use comma “,” without spaces as separator.

/Schema** <http://standards.iso.org/iso/17442>

By default, it is the LEI schema: <http://standards.iso.org/iso/17442>

Schema attribute related with the identifier of the **/**Entity** (see above). Usually given by the regulator. ESEF mandates the LEI schema, therefore used as default.

See more at codes.eurofiling.info.

Parameters for taxonomy extension only.

/Label** http://xbrl.ifrs.org/taxonomy/2016-03-31/full_ifrs/labels/lab_full_ifrs-en_2016-03-31.xml

/Label** http://xbrl.ifrs.org/taxonomy/2016-03-31/full_ifrs/labels/lab_full_ifrs-es_2016-03-31.xml

The specific language “**xx**” is detected as two consecutive letters placed between whatever separators (- _ . /) when matching an ISO 639-1 code.

Note this parameter must be repeated for each language “**xx**” in the base taxonomy. The labels for such language “**xx**” in the base taxonomy will be found here, accordingly. Additional labels optionally may be declared for such language in RECEiX dictionary (column **Lang=“xx”**).

/Extension** <http://www.openfiling.info/cbso-rf/2018-01-01>

Used for taxonomy extension only.

The calculated parameters for the taxonomy extension are all derivatives of the declared value, whose components are shown below:

extension	http://www.openfiling.info/cbso-rf/2018-01-01
extensionXmlns	cbso-rf
extensionHref	cbso-rf_2018-01-01.xsd
extensionDirectory	www.openfiling.info
extensionItemIdPrefix	cbso-rf_
extensionDate	2018-01-01

Additional optional parameters, for expert use only.

All the information required for the iXBRL instance document are obtained from the previous parameters. However, XBRL experts may add text declarations, to be concatenated at the end of the corresponding iXBRL containers, as follows:

/Head**

Optional. Includes the content in column Fact inside the XHTML tag `<head>...</head>`

Note RECEiX generates by default all the required `<head>...</head>` stuff.

/Resources**

Optional. Includes the content in column Fact inside the iXBRL tag `<ix:resources>...</ix:resources>`

Note RECEiX generates by default all the required `<ix:resources>...</ix:resources>` stuff

/References**

Optional. Includes content in column Fact inside the iXBRL tag `<ix:references>...</ix:references>`

Note RECEiX generates by default the `<ix:references>...</ix:references>` stuff.

/Hidden**

Optional. Includes the content in column Fact inside the iXBRL tag `<ix:hidden>...</ix:hidden>`

Note RECEiX generates by default all the required `<ix:hidden>...</ix:hidden>` stuff.

/Tail**

Optional. Includes content in column Fact after the end of the iXBRL document.

Note RECEiX generates by default all the required stuff at the end of the iXBRL document.

Footnotes spreadsheets (optional).

The RECEiX arrangements for declaring footnotes follow the customary uses in financial statements.

Footnotes, if used, must be placed in one or several independent spreadsheets, with the Excel spreadsheet name starting with the letters **“not”** (in upper or lower case or combinations).

Examples are a single spreadsheet with the name of **“Notes”** or several spreadsheets named **“Notes English”**, **“Notas español”** and so on.

If the Excel spreadsheet name includes the string **“hidden”**, as in **“Notes-hidden”**, then the footnotes will be generated (according to the iXBRL specification) in an iXBRL hidden area, not visible in web browsers.

Note that a *footnote* spreadsheet can also contains other iXBRL facts.

Each footnote is composed of several consecutive rows, as follows:

The first row of a footnote is denoted for:

(master type) containing a cell that is referenced (hyperlinked) from another cell in a row of an **“entry”** spreadsheet which also contains an iXBRL fact. The cell referencing the note must have an iXBRL cell in the same row and at its right (in a higher column). For example, the row 10 of the spreadsheet **“Assets”** has a cell **‘Assets’!D10** with an hyperlink referencing the cell **‘Notes’!A4** and an iXBRL fact in the cell **‘Assets’!H10**. Therefore, the referenced footnote starts in the row 4 of the **‘Notes’** spreadsheet. The footnote is linked in iXBRL to the first iXBRL fact at its right in the same row, in this example to the fact in the cell **‘Assets’!H10**

(translation type) having a cell which reference (hyperlinks) to a **“master type”** footnote. This **“translation type”** footnote heritages all the iXBRL links from the **“master type”** footnote. In the example, the cell **‘Notes’!A7** has a hyperlink referencing the cell **‘Notes’!A4**. This footnote applies by heritage also to the cell **‘Assets’!H10**

Language: If a two letters ISO 639-1 code is placed in any cell of the first row of a footnote, the first ISO 639-1 code found will be the `xml:lang` used for the footnote. Otherwise, the default language for the iXBRL instance document (see `Lang=“xx”` in *data* spreadsheet) will be used. In the example, language would be explicitly declared as **“es”** in the cell **‘Notes’!B4**, or if left in blank the default language (usually **“en”**) is assumed.

Rows inside a footnote:

The iXBRL footnote includes the first row of the footnote, the last row of the footnote, and all the intermediate rows. The iXBRL footnote content will be represented as an *.html* table.

Last row of a footnote:

Is the row just prior the following footnote or the last row of the *footnotes* spreadsheet

Errors sheet.

The errors detected and relevant alerts are displayed with a message in the screen. For facilitating the error handling, a spreadsheet may be declared as container for errors.

Simply create spreadsheet starting with the letters “err” (upper or lower case mixed), and each error will be also placed in a consecutive row of this “Error” spreadsheet, displaying the alert or the error detected, as well as the offending spreadsheet and row (if found).

Only the first spreadsheet starting with the letters “err” is considered. The others (if any) are ignored and not processed at all.

Exported Path and File name

A cell in the Workbook must have the name “FilePath”. This cell is used to set the path and the filename of the exported iXBRL file (and also the XBRL file, if exists). The default path and file name are the same path and file name of the workbook.



Favicon (optional)

The [favicon](#) is a small image appearing in the Navigator tab of an *xhtml* file, as follows:



If (and only if) a merged cell in the Workbook has the name “Favicon”, and if this merged cell contains an image, then the image will be used as favicon in the Navigator tab. No default is available.

Validator (optional)

If (and only if) a cell in the Workbook has the name “Validator”, then this cell should contain the path to the Validator program.

Type “SKIP” instead a path the cell name “Validator” to inhibit (skip) validation.

RECEiX uses Arelle as validator. Arelle is an XBRL Certified Open Source suite for XBRL, which includes validation features. RECEiX will invoke Arelle as XBRL validator for the exported iXBRL and/or XBRL file, and the results of the validation will be show in the “Errors” sheet. Feel free to download Arelle from www.arelle.org The default path is C:\Program Files\ArELLE



Taxonomy extension (optional).

This step creates an *extension* taxonomy. The starting point are the *base* (source) taxonomy items (a subset of IFRS 2016 in the example) plus a number of (additional / new) extended taxonomy items (in the example ERICA proprietary items), being each item described in a row of the “*dictionary*” spreadsheet. Each property of the item is placed in a specific column of the row.

The meaning of each specific column is identified by a string value (column short description) placed in the header row (row number 7). The first empty row after the header row denotes the end of the “*dictionary*” spreadsheet

The two characters */** placed in column 1 of a row denotes the row is not to be processed as item row (skipped). This arrangement is useful for placing comments.

Note: This step is optional. RECEiX can also be used for generate iXBRL instance document without generating the corresponding extension taxonomy. Simply delete this “*dictionary*” spreadsheet, hide it, or prefix its name with the string “*hidden*”³.

Taxonomy extension directories and files

In the directory in which RECEiX is executed, a *Taxonomy Package* directory will be created following the XII Working Group Note 14 August 2018 *Including XBRL Reports in Taxonomy Packages 1.0* as published at www.xbrl.org/WGN/report-packages/WGN-2018-08-14/report-packages-WGN-2018-08-14.html

The *Taxonomy Package* directory will take (heritage) the same name than the RECEiX file name, but without the file type extension *.html*

Inside the *Taxonomy Package* directory will be created:

- *.html* file containing the iXBRL instance document, with the same file name than the RECEiX file name, but with the file type extension *.html* instead of *.xslm*. Note that the directory for this file is different when RECEiX extends the taxonomy, due to the use of Taxonomy Package specification.
- *extensionDirectory* directory, named as derivate from */**Extension* parameter. The *extensionDirectory* will be populated by RECEiX with other two nested directories inside: *extensionXmIns* for this type of taxonomy and yet inside *extensionDate* containing this specific taxonomy extension. E.g.: www.openfiling.info/cbso-rf/2018-01-01.
- *META-INF* directory containing other required Taxonomy Package files, according to XBRL Taxonomy Packages specification as in <https://specifications.xbrl.org/spec-group-index-taxonomy-packages.html>

³ For instance, changing the spreadsheet name as *hiddendictionary* instead of *dictionary*. The spreadsheet will be ignored as *hidden* without affecting any XBRL facts.

Columns in the dictionary spreadsheet

RECEiX requires the declaration in the “*dictionary*” spreadsheet of the following attribute columns, consecutively (no intermediate columns in blank), but in whatever order, to be placed in the row 7:

Qname

Is the Qname of the item. E.g.: *ifrs-full:StatementOfFinancialPositionAbstract*

Note the separator “:” indicating at the left the *xmlns* descriptor (e.g.: *ifrs-full*) and at the right the *tag* descriptor (e.g.: *StatementOfFinancialPositionAbstract*). Only two *xmlns* descriptors are allowed: one for the base taxonomy (e.g.: *ifrs-full*) and the other for the extension taxonomy (e.g.: *csbo-rf*).

The *tag* descriptor would be hence duplicated without confusion; both in the base and the extension taxonomy. Note this practice may be prone to human errors.

A common practice among XBRL practitioners is defining the *tag* descriptor using Label Camel Case Concatenation (LC3) [use case for XBRL](#) and *in a language customary in the sphere of international finance* (euphemism for English).

Lang="en"

Label (optional) for the Item in the language “en”. E.g. *GENERAL CHARACTERISTICS*

Insert a different column **Lang="xx"** (consecutive or not) for the labels in each required [ISO 639-1](#) language. E.g. **Lang="en"** plus **Lang="es"** plus **lang="ru"** plus **lang="ja"**

A useful style guide for labels is public at xbrl.us/xbrl-reference/style-guide/

The optional parameter */**Label* (see *Parameters* chapter) is used for each (different language) label linkbase of the base taxonomy Items.

For a **Lang="xx"** column without the corresponding */**Label* parameter declared for this language “xx”, each label optionally defined in the **Lang="xx"** column will be created in the label linkbase using the xlink:role *label* only, irrespective if it is a base taxonomy Item or an extension taxonomy Item.

For a **Lang="xx"** column with the corresponding */**Label* parameter declared for the same language “xx”, each label optionally defined in the **Lang="xx"** column will be created in the label linkbase using the xlink:role *terseLabel*, irrespective if it is a base taxonomy Item or an extension taxonomy Item. Additionally, if it is an extension taxonomy Item, a xlink:role *label* will be also created. Note the xlink:role *label* has been already declared for each base taxonomy Item in the */**Label* parameter

At least one **Lang="xx"** column must be declared. If a **Lang="en"** column is declared, then “en” is the default language. If no **Lang="en"** column is declared, then the default language is the language of the first **Lang="xx"** column declared.

For the default language, if a label is not declared for (a) an extended taxonomy Item or (b) for a base taxonomy Item when the */**Label* parameter is missing for such language, then RECEiX will place in the accordingly a default label in the dictionary based in the *tag* descriptor interpreted as a Label Camel Case Concatenation. See details at www.xbrl.org/technical/guidance/FRTA-RECOMMENDATION-2005-04-25.htm# 2.1.4

ItemType

ItemType declaration is mandatory for all items, both base taxonomy items (for checking purposes) and extension taxonomy items. Please check that the ItemType of the base taxonomy items are accurately copied here.

[Abstract|Text|TextBlock|String|Monetary|Date|Integer|Pure|Shares|Percent|PerShare] ItemTypes are used for the definition of iXBRL facts. All of the items of these types (except [Abstract] type) can be used in the *data* spreadsheet.

Note [Text] ItemTypes is an alias of [TextBlock]

[Table|Dimension|Domain|Member|LinItem] ItemTypes are used for dimensional declarations. Only the [Dimension|Domain|Member] ItemTypes can be used in the *data* spreadsheet and only in the columns for dimensions.

Restriction: Other iXBRL ItemTypes are not yet implemented.

Note: The first item declared in the *directory* spreadsheet must be [Text] and being an extension taxonomy item. This first item is used as the QName for the linkroleId being the value on the Parent column a free text describing the linkrole.

Period [Instant|Duration].

Mandatory for extension taxonomy items. Let in blank for base taxonomy items.

Balance [Credit|Debit].

Mandatory for extension taxonomy [Monetary] items. Let in blank otherwise

Weight [1|-1]

Optional use, and only for [Monetary] items, for both base and extension taxonomy items. This attribute applies on the XBRL calculation linkbase checks.

Anchor

The Anchor QName of an Item QName is another Item QName having both Qnames a relation wider-narrower as described in [ESEF Reporting Manual](#)

Anchoring may used only for Numeric Qnames with empty Weight. In all the other cases Anchoring is automatically erased.

If the Anchor of a candidate QName is empty, then RECEIX try to place Parent QName as default anchor when (1) QName and Parent QName are both Numeric and (2) Weight is empty and (3) QName and Parent QName are declared one in the base taxonomy and the other in the extended taxonomy.

Note Anchoring specification is still an XBRL International Working Group draft

Calculation and Presentation: Parent

The Parent QName of an item QName is another item QName one level above in the hierarchy. E.g. Item [ifrs-full:CurrentLiabilities] with Parent [ifrs-full:Liabilities]

A Item may have two different parent Qnames, one used for calculation (when attribute Weight is declared) and the other used for presentation, with the following different rules.

Calculation Parent - Children Rules:

If the attribute [Weight] is declared, then the value of the Parent must match with the summation \sum of the value of each child multiplied by their respective Weight [1|-1].

E.g. $\text{ifrs-full:Assets} = (1 * \text{ifrs-full:CurrentAssets}) + (1 * \text{ifrs-full:NoncurrentAssets})$.

Note that circular⁴ summations \sum are invalid and will be detected by RECEiX.

Note also that the ItemType of the Parent, if declared, must be Numeric.

Presentation Parent - Children Rules:

The presentation order of parents and their respective children follows the order of their respective declarations in the *dictionary spreadsheet* and must result in a tree structure, as defined in the article en.wikipedia.org/wiki/Tree_structure.

The tree structure depends on the order of declaration of Parent Items in the *dictionary spreadsheet* as well as the Parent of each Item.

The first Item placed in the *dictionary spreadsheet* has not Parent (root node).

If the Tag declaration of the Parent does not end with the string “*abstract*”, and other Item with the same Tag declaration of the Parent but ending with the string “*abstract*” appears before in the directory (irrespective if belonging to the base of extended taxonomy), then the “*abstract*” Parent is used instead.

e.g. if *ifrs-full:AssetsAbstract* is declared prior to *ifrs-full:Assets* in the directory, then *ifrs-full:AssetsAbstract* will be used as presentation Parent for all the children of *ifrs-full:Assets*.

Even more, if *cbso-fr:AssetsAbstract* is declared prior to both *ifrs-full:AssetsAbstract* and *ifrs-full:Assets* in the directory, then *cbso-fr:AssetsAbstract* will be used as presentation Parent for all the children of *ifrs-full:Assets*.

This kind of arrangement with a first Tag declaration Item ending in “*abstract*” for presentation and the same Tag declaration not ending in “*abstract*” for calculation is customary in IFRS.

Note that RECEiX will determine the presentation structure in the best possible way according to the above rules, raising an error when the tree structure cannot be properly determined.

⁴ I.e. $A=B+C$ and $C=A+B$. The result of a summation must not be a direct or indirect component of itself.

Dimension: Parent

Used for the following Item Types: [Table | Dimension | Domain | Member | Lineltem]
constructing the dimensional declaration of tables as follows:

Table

Mandatory row at the beginning of each table definition.

Creates a specific definitionLink for this Table.

A Table must have at least a Dimension and at least a Line Item.

The value of the Parent of this element is free text used for the linkrole

The QName must end in "Abstract". Otherwise, the suffix "Abstract" will be added

The QName will be generated three times, with a suffix of "Abstract" for the full table, with a suffix of "Table" for the Hypercubes, and with a suffix of "Lineltems" for the Items.

Dimension

Mandatory row at the beginning of each Dimension.

Must have a Table as Parent. If a Dimension is used in several tables, simply define a row for each use of Dimension/Table.

A Dimension may have an optional Domain and must have at least a Member.

Dimension can be used in *data* spreadsheet, in the Dimension column of the Dimension DimensionMember pair of columns.

Note only explicit Dimensions can be declared. Typed Dimensions are not implemented yet.

Domain.

Optional. Must have a Dimension as Parent.

When declared, it is also used as *dimension-default* (ESEF requires *dimension defaults*)

Domain can be used also as a Member in *data* spreadsheet.

Member.

The Parent must be a Dimension, a Domain or another Member

Member can be used in *data* spreadsheet, in the DimensionMember column of the Dimension DimensionMember pair of columns.

Lineltem

Container for facts. The Parent must be a Table or other Lineltem

The Lineltem must have a twin brother (with the same QName) declared in other row as [Text | TextBlock | String | Monetary | Date | Integer | Pure | Shares | Percent | PerShare]

Presentation, Indent, NestLevel optional columns:

RECEiX populates the output data resulting of the process in the following columns (provided the columns is declared).

Presentation

The Parent QName selected for the presentation of this Item. E.g.: ifrs-full:PropertyPlantAndEquipmentAbstract as presentation Parent of ifrs-full:LandAndBuildings

Indent

Numerical expansion of the tree structure.

E.g.: .1 .1.1 .1.2 .1.2.1 .1.2.2 .1.2.3

NestLevel

Number of the intermediate nodes between this Item QName and the root Item QName (Nest Level 1). E.g.: (in reference to prior indent example) 2 3 3 4 4 4