

# Revenue



Cáin agus Custaim na hÉireann  
Irish Tax and Customs

---

## Electronic Filing of Financial Statements Inline XBRL (iXBRL)

### Style Guide

Document Version	2024.0 (replaces 2023.0)
Document release date	December 2024



## Contents

1. INTRODUCTION.....	4
1.1 What's New?.....	4
2. GENERAL GUIDANCE.....	4
2.1 Conformance with the Inline XBRL Specification.....	4
2.2 Document Creator / Product Information .....	4
3. DOCUMENT LEVEL ISSUES.....	4
3.1 Input Document .....	4
3.2 File Type .....	5
3.3 Single Output Document.....	5
3.4 XHTML vs HTML .....	5
3.5 Document Encoding.....	5
3.6 Use of “non-standard” Characters.....	6
3.7 CSS Styling .....	6
3.8 Java Applets, JavaScript .....	7
3.9 Pagination .....	7
3.10 Images, logos, etc.....	7
3.11 XML Canonicalisation & Encoding Submissions.....	7
3.12 Hidden data items.....	8
3.13 Entity Identifier Scheme.....	8
4. ELEMENT-LEVEL ISSUES .....	9
4.1 Transformation rules .....	9
4.2 Escaped HTML.....	9
4.3 Number Signs .....	9
4.4 Scaling and Precision.....	10

---

4.4.1	<i>Inconsistent Duplicate Facts</i> .....	11
4.5	Percentages.....	11
4.6	Nesting Mark-up .....	11
4.7	Marking up Boolean items.....	12
4.8	Marking up empty “flag” items.....	12
5	FURTHER INFORMATION.....	13

## 1. INTRODUCTION

Since 23 November 2012, financial statements in iXBRL format have been accepted via the Revenue On-Line Service (ROS). All Income Tax and Corporation Tax payers now have the option of uploading their financial statements.

For Corporation Tax payers, the obligation to file iXBRL financial statements is being conducted in phases and details of who is obliged to file and when can be found in Paragraph 2.1 the [iXBRL Tax and Duty Manual](#).

This document is aimed at software developers and those who are using conversion tools to prepare accounts in iXBRL. The guide lays out rules and guidance on the preparation of inline XBRL documents for subsequent submission to Revenue. It does not cover information relating to the taxonomies accepted by Revenue or the mechanism for online filing – for these please refer to further documentation available at

<https://www.revenue.ie/en/companies-and-charities/corporation-tax-for-companies/submitting-financial-statements/index.aspx>

### 1.1 *What's New?*

General edit of 'Overview' to remove specific references to Revenue's former 'Large Cases Division' and instead refer readers to Para 2.1 of the [iXBRL Tax and Duty Manual](#) for information on who is obliged to file iXBRL financial statements and when.

## 2. GENERAL GUIDANCE

### 2.1 *Conformance with the Inline XBRL Specification*

The service is aligned with the Inline XBRL Specifications, versions 1.0 and 1.1, which can be found at:

<https://specifications.xbrl.org/spec-group-index-inline-xbrl.html>

### 2.2 *Document Creator / Product Information*

**FRS 101 + DPL, FRS 102 + DPL and EU IFRS + DPL taxonomies**

Please use the NameProductionSoftware tag to identify the software package and the VersionProductionSoftware tag to identify the version of the software package.

## 3. DOCUMENT LEVEL ISSUES

### 3.1 *Input Document*

Financial Statements must be comprised of a single, self-contained Inline XBRL document that can be opened and viewed in a single browser window. Multiple Inline XBRL documents are not permitted by the service.

A maximum file size limit is applied to iXBRL submissions. Filers should keep the file size of their submissions as small as possible in order to ensure acceptance and facilitate prompt validation.

### 3.2 File Type

The list of acceptable file extensions for upload is: html, htm, ixbrl, xml, xhtml.

### 3.3 Single Output Document

The result of processing a submitted Inline XBRL document is always a single target XBRL instance document. Multiple target XBRL instance documents are not supported by the service, despite being permitted by the Inline XBRL Specification.

### 3.4 XHTML vs HTML

While the Inline XBRL Specification allows for the use of either HTML or XHTML, documents submitted via Revenue services must be well-formed XML and therefore valid XHTML. The <html> root element of your Inline XBRL document should be in the XHTML namespace. This is usually achieved by setting the default namespace for the document, e.g.

```
xmlns='http://www.w3.org/1999/xhtml'
```

All other namespaces required for Inline XBRL should also be included.

A DOCTYPE declaration for Inline XBRL documents must be omitted.

The XHTML version attribute can be used to identify a document as Inline XBRL, but its use is optional. If used, the value must be the Formal Public Identifier “-//XBRL International//DTD XHTML Inline XBRL 1.0//EN”

For example:

```
<html
  xmlns='http://www.w3.org/1999/xhtml'
  xmlns:iso4217='http://www.xbrl.org/2003/iso4217'
  xmlns:ix='http://www.xbrl.org/2008/inlineXBRL'
  xmlns:link='http://www.xbrl.org/2003/linkbase'
  xmlns:xbrli='http://www.xbrl.org/2003/instance'
  xmlns:xlink='http://www.w3.org/1999/xlink'
  xmlns:ixt='http://www.xbrl.org/inlineXBRL/transformation/2010-04-20'
  xmlns:uk-gaap='http://www.xbrl.org/uk/gaap/core/2009-09-01'
  xmlns:uk-bus='http://www.xbrl.org/uk/cd/business/2009-09-01'
  xmlns:uk-direp='http://www.xbrl.org/uk/reports/direp/2009-09-01'
  xmlns:uk-aurep='http://www.xbrl.org/uk/reports/aurep/2009-09-01'
  xmlns:uk-countries='http://www.xbrl.org/uk/cd/countries/2009-09-01'
  xmlns:ie-common='http://www.xbrl-ie.net/common/core/2012-12-01'
  xmlns:ie-gaap='http://www.xbrl-ie.net/gaap/core/2012-12-01'
  xmlns:xbrldi='http://xbrl.org/2006/xbrldi'>
```

### 3.5 Document Encoding

iXBRL documents submitted to Revenue should be UTF-8 encoded by default.

The MIME type selected for the Inline XBRL Document will often have an impact on whether browsers display the document correctly. The MIME type can be set within the Inline XBRL Document with code such as:

```
<meta content='text/html; charset=utf-8' http-equiv='Content-Type' />
```

Authors of Inline XBRL Documents should test how their documents display in different browsers to ensure that the MIME type is set appropriately.

When submitting an Inline XBRL document to Revenue via ROS web-services the iXBRL document is embedded within an envelope/xml document. As such the XML declaration (`<?xml/ version="1.0" encoding="utf-8"?>`) and any leading Byte-Order Mark must be removed. The complete XML request should have an initial XML declaration.

### 3.6 Use of “non-standard” Characters

There are two principal methods of including “non-standard” characters (characters that fall outside of the ASCII 7-bit character set) in inline XBRL documents (such characters may include currency symbols, simple graphic symbols, accented characters, etc). The first is to use the appropriate UTF-8 encoding, which will usually (but not always) be a two-byte value, both of which will have their top (i.e. 8th) bit set. So long as the document encoding is correctly set then browsers or other display tools will be able to interpret these bytes correctly as the appropriate character. However, be aware that text editors and other display tools that don’t understand document encodings and treat the content as 7-bit ASCII will not display UTF-8 encoded characters correctly, and if used to modify a document with embedded UTF-8 encoded characters may actually destroy or alter the encoding such that it no longer renders correctly in UTF-8-aware applications.

The second method may be employed to avoid such eventualities. XML (of which XHTML is an application) provides for escaped character references of the form “&#xxxx;” where “xxxx” may be replaced by a decimal number representing the Unicode code point (see [https://en.wikipedia.org/wiki/Code\\_point](https://en.wikipedia.org/wiki/Code_point) for more details) of the character glyph to be displayed. For example, here are some common symbol glyphs and their character reference equivalents:

£ ( <i>pound sign</i> )	&#163;
€ ( <i>euro sign</i> )	&#8364;
© ( <i>copyright</i> )	&#169;

Such code point values are also encoded into the bytes of a UTF-8 encoded multi-byte character – you can think of an XML character reference and a UTF-8 encoded character as two alternative ways of specifying a Unicode code point, which in turn represents a particular character glyph.

The advantage of the second method is that non-UTF-8-aware text editors and text display tools will not disrupt or damage the character references, though they will not be displayed as intended.

The exceptions to the rule are the five characters special to XML mark-up: ‘<’, ‘>’, ‘&’, ‘”’ and ‘”’. Their ASCII character representations are UTF-8 compatible, so to protect them from interpretation by XML-aware processors, XML provides five named character entities to escape them (“&lt;”, “&gt;”, “&amp;”, “&apos;” and “&quot;”). You can use these named character entities in preference to the five equivalent numeric character references if you wish.

It is recommended that you avoid using named character entities aside from these five “special” ones. Named character entities are usually defined in a DTD (Document Type Definition) and the XHTML DTD defines quite a number of them. However, inline XBRL does not have a fully defined DTD (the preference being for the modular XHTML Schema) meaning that the interpretation of these named entities becomes browser-dependent.

### 3.7 CSS Styling

Cascading Style Sheet (CSS) language is the mechanism that should be used to apply styling information. If CSS is used it must be embedded in the Inline XBRL document directly. It is not possible to provide a separate (.css) file for inclusion.

### 3.8 Java Applets, JavaScript

Java applets, JavaScript, and any other HTML <script> content, is embedded code executed by a browser or other rendering engine, and presents an unacceptable security risk for Revenue internal systems. It is therefore not permitted to submit Inline XBRL documents with embedded applets or script fragments of any kind either to animate the document or enhance the look & feel of the document. This is best achieved with CSS.

### 3.9 Pagination

Page breaks should be inserted at appropriate points in the Inline XBRL document to ensure that printed output looks as neat and as presentable as possible. Page breaks should be inserted before major headings, tables, front-sheets, etc. to ensure that, as much as possible, important information isn't spread over a physical page boundary. These breaks should mirror the page breaks inserted by generating applications when producing local renderings in other formats (e.g. plain text, PDF, Word).

In XHTML, "page break before" and "page break after" styles can be applied to mark-up elements directly. For example:

```
<p style="page-break-after: always">.....</p>
<table style="page-break-before: always">.....</table>
```

### 3.10 Images, logos, etc

Any logos or small images that are integral to the look & feel of a submitted Inline XBRL document should be inlined as a base64-encoded string in the URI value normally used to reference an image file using the 'data' URI scheme. E.g.:

```

```

Where 'xxx...xxx' is the base64-encoding of the image whose mime-type is 'image/png' (GIF, JPG and PNG are the only widely supported image formats). This technique is manageable with relatively small images, logos, etc., but should be used sparingly to avoid bloating the overall Inline XBRL document. If an image is required in more than one location it is possible to create a CSS macro utilising the background:url style and passing it the inlined URI. E.g. (assuming a 100x100 pixel image):

```
DIV.img {width:100px;
height:100px;

background:url(...xxx);}
```

### 3.11 XML Canonicalisation & Encoding Submissions

Those opting to use ROS web-services for iXBRL submission must digitally sign the SOAP message body, including the iXBRL document contents.

The raw XML (XHTML) can be included within the <InlineXbrl> element or alternatively a Base64-encoded string version of the file contents can be included within the <EncodedInlineXbrl> element. The Base64 option may be preferable for applications seeking to avoid generating a canonical version of the Inline XBRL contents.

The Revenue service treats base-64 encoded Inline XBRL documents identically to “ordinary” documents with the exception of an additional decoding step. Take care not to include any Byte-Order Mark (BOM) that may be present when encoding. Its removal is the responsibility of the encoding/filing application.

### 3.12 *Hidden data items*

Inline XBRL provides a mechanism (*ix:hidden*) for marking up data items that do not appear on the face of a document. This mechanism should be used as sparingly as possible – the more data that is “hidden” the harder it is to assure that the human and computer readable data items in the document are aligned – when they are one and the same value, alignment is assured.

The *ix:hidden* section of an Inline XBRL document should be reserved for:

- The declaration of tuple container elements
- Document metadata, including creator information
- Boolean items not directly associated with statements on the face of the document
- “Flag” items not directly associated with facts on the face of the document
- Facts that do not normally appear on the face of the document

Furthermore, the use of HTML or CSS text hiding mechanisms other than that conventionally used to protect the *ix:hidden* section from view are strongly discouraged. If in doubt, follow the general principle that if a fact appears on the face of a document, then it should be marked-up there and not by any method that involves the hiding of data and mark-up.

### 3.13 *Entity Identifier Scheme*

Note that the scheme URI should appear with a trailing ‘/’ For example,

```
<xbri:entity>
```

```
<xbri:identifier
```

```
scheme="http://www.revenue.ie/">1234567T</xbri:identifier>
```

```
</xbri:entity>
```



## 4. ELEMENT-LEVEL ISSUES

### 4.1 Transformation rules

The Inline XBRL Transformation Rules Registries define all the transformation rules that are applied to displayed fact values in an inline XBRL document to turn them into canonical XSD-type representations suitable for inclusion in the resulting XBRL target document.

The transformation namespaces currently accepted by Revenue are:

<http://www.xbrl.org/inlineXBRL/transformation/2010-04-20>

<http://www.xbrl.org/inlineXBRL/transformation/2011-07-31>

<http://www.xbrl.org/inlineXBRL/transformation/2015-02-26>

Please note that the Revenue system allows for any combination of registries provided that the namespace(s) are declared correctly and the transformations used exist within the declared transformation registry.

### 4.2 Escaped HTML

The Inline XBRL Specification allows textual data in a *ix:nonNumeric* marked-up fact to contain ‘escaped’ (X)HTML formatting. This is not a feature of the Specification that will be supported by Revenue’s service. (X)HTML mark-up in any other Inline XBRL element is not permitted.

### 4.3 Number Signs

Numbers represented in XBRL (and Inline XBRL) are generally positive in value, notwithstanding the value of the ‘balance’ attribute in the concept’s definition, which is usually of use only in calculation relationships. Where a genuine negative value is required in the target XBRL document (e.g. to show a loss for a concept that is defined in terms of profit) the ‘sign’ attribute in the Inline XBRL mark-up should be used to indicate this.

The representation of the sign in the Inline XBRL document rendering is completely separate and is a function of the XHTML mark-up that surrounds the marked-up fact. That mark-up might indicate a negative value with surrounding brackets, or by a leading minus sign, or by changing the font to red (or a combination of these) – the point is that none of these adornments, by themselves, have any influence over the sign of the value that ends up in the target XBRL document, which by default will be positive unless the sign attribute is used as an indicator.

The following examples should illustrate the differences:

#### **a. Negative underlying value with negative portrayal**

The monetary fact representing a loss in a statement of the form:

Profit (Loss) before Tax (345,678)

in the Inline XBRL rendering of a set of Accounts might be marked up in the inline XBRL document as a table data item as follows:

```
<td><ix:nonFraction name="PBT" unitRef="eur"
sign="-"
format="ixt:numcommadot">345,678</ix:nonFraction
></td>
```

Note that the brackets are outside the Inline XBRL mark-up so as not to interfere with the value that is transformed for the target XBRL instance document. The presence of the sign attribute results in a negative value in the resulting XBRL instance document that correctly indicates that the value represents a loss rather than a profit: <PBT unitRef="eur">-345678</PBT>.

The item PBT would normally be defined in the Taxonomy with its balance set to 'credit' to indicate that where a profit is reported, it is naturally thought of as an increase in funds (i.e. a credit), and so a loss must be explicitly negated.

**b. Positive underlying value with negative portrayal**

The monetary fact representing cost of sales in a statement of the form:

Cost of Sales	(123,456)
---------------	-----------

in the inline XBRL rendering of a set of Accounts might be marked up in the Inline XBRL document as a table data item as follows:

```
<td><ix:nonFraction name="CostSales" unitRef="eur"
format="ixt:numcommadot">123,456</ix:nonFraction></td>
```

Note that the brackets are outside the Inline XBRL mark-up so as not to interfere with the monetary value that is transformed for the target XBRL instance document.

However, in this case, the lack of a sign attribute results in a positive value in the resulting XBRL instance document that correctly indicates that the value represents a cost amount:

```
<CostSales unitRef="eur">123456</CostSales>
```

The item CostSales would normally be defined in the Taxonomy with its balance set to 'debit' to indicate that where a cost is reported it is naturally thought of as a decrease in funds (i.e. a debit) and so doesn't need to be explicitly negated.

**c. Negative underlying value with positive portrayal**

The monetary fact representing a cost of disposal in a statement of the form:

Disposal Cost	45,678
---------------	--------

in the Inline XBRL rendering of a set of Accounts might be marked up in the Inline XBRL document as a table data item using a revenue tag as follows:

```
<td><ix:nonFraction name="DisposalRevenue" unitRef="eur" sign="-"
format="ixt:numcommadot">45,678</ix:nonFraction></td>
```

Note that in this (slightly contrived!) example the cost is being tagged as a negative revenue. The presence of the sign attribute results in a negative value in the resulting XBRL instance document that correctly indicates that the value represents a cost amount:

```
<DisposalRevenue unitRef="eur">-45678</DisposalRevenue>
```

The item DisposalRevenue would normally be defined in the Taxonomy with its balance set to 'credit' to indicate that where revenue is reported it is naturally thought of as an increase in funds (i.e. a credit) and so in this case needs to be explicitly negated to show a cost.

## 4.4 Scaling and Precision

Numbers may be conveniently scaled in human-readable reports to show, for instance, figures in round thousands or millions. When marking up financial reports it is necessary to ensure that if scaling is required the correct precision is declared and the correct rounding is applied to the value to ensure

that all occurrences of the same fact (whether appearing scaled or not) remain equivalent within the limits of their declared precisions.

For instance, an underlying value such as €1,234,567.89, which is accurate to 2 decimal places, may appear rounded to the nearest Euro with a decimals value of '0' (i.e. €1,234,568), rounded to the nearest thousand with a decimals value of '-3' and a scale value of '3' (i.e. €1,235), or rounded to the nearest tenth of a million with a decimals value of '-5' and a scale value of '6' (i.e. €1.2). The latter, when un-scaled and converted to canonical form for XBRL would result in a value of €1,200,000 (accurate to -5 decimal places), which nevertheless is "equivalent" to €1,234,567.89 (accurate to 2 decimal places) within their stated accuracies. However, any 'higher' value for decimals (e.g. '-4') would result in an inconsistency.

For example,

```
<td><ix:nonFraction name="DisposalRevenue" unitRef="eur" decimals="-5"
scale="6" format="ixt:numcommadot">1,235</ix:nonFraction></td>
```

#### 4.4.1 Inconsistent Duplicate Facts

If rounding has been misapplied to one or more occurrences of duplicate values in the original document (as may occur when summarising more accurate financial data from the body of a report) then the duplicate values may be inconsistent within the bounds of their stated accuracies. Such inconsistencies will result in submission rejection.

### 4.5 Percentages

Numbers which are percentages should typically be entered in Inline XBRL with scale=-2. The number 0.54 represents a percentage of 54%. 1 represents 100%.

The "decimals" attribute representing the accuracy of the figure will typically be 2, meaning that 54%, or 0.54, is accurate to at least that number of figures. However, it may be more than 2 and can range up to infinity if the percentage is exact (as it might be, for example, with a tax rate).

A unit based on 'pure' should be applied to any percentage values. Note that the percent sign itself occurs outside of the Inline XBRL mark up.

For example,

```
<td><ix:nonFraction name="Year1Rate" contextref="AP1" unitref="PureUnit"
decimals="2" scale="-2">54</ix:nonFraction>%</td>
```

Most tags representing percentages are percent data types, but some are decimal data types. This makes no difference to the form of data entry.

### 4.6 Nesting Mark-up

The Inline XBRL Specification allows the nonNumeric element to contain other inline XBRL mark-up so that where narrative text is marked up as a single block of text within a single string-type item using ix:nonNumeric it is still possible to mark-up embedded facts so that they appear as discrete items in the target XBRL instance document, in addition to the item representing the text block as a whole. This is particularly useful where a value or a significant name is mentioned in the middle of a paragraph of text (such as the Director's Report or one of the Notes to the Accounts).

For example, the description of a contingent liability with an embedded current period liability figure might be marked up as follows:

```
<td><ix:nonNumeric name="DescriptionLiability" contextref="CY">The company operates a
confidential invoice discount facility. The balance due on this facility was €<ix:nonFraction
```

```
name="AmountLiability" contextref="CY" unitref="EUR"
format="ix:numcommadot">5,000</ix:nonFraction> at 31st March 2010.</ix:nonNumeric></td>
```

#### 4.7 Marking up Boolean items

Version 2 of the transformation registry includes two new transformations, *ix:booleanfalse* and *ix:booleantrue*, which allow for arbitrary text to be marked up as normal but for the resulting value to be either “true” or “false” in the target XBRL instance.

Version 2 of the transformation registration is now supported, but if you are using Version 1, the *ix:exclude* element can be used to simulate this behaviour as follows;

If we assume that the sentence “The company is trading.” appears on the face of the document then the following “pattern” will allow the Inline XBRL mark-up of this text with a Boolean item such as *uk-bus:EntityTrading*:

```
<ix:nonNumeric name="uk-bus:EntityTrading contextRef="CY">
    <ix:exclude>The company is trading.</ix:exclude>
    <div style="display:none">true</div>
</ix:nonNumeric>
```

The nested *ix:exclude* element allows part of the content within the *ix:nonNumeric* element (the statement itself) to be excluded from the item in the target XBRL instance document, whilst the CSS style on the *div* element suppresses display of the actual Boolean value on the face of the document, which is the part of the marked-up value that is carried into the target XBRL.

#### 4.8 Marking up empty “flag” items

There are a handful of concepts (of type *fixedItemType*) in the Accounts Taxonomies whose presence in the target XBRL instance document is sufficient to convey a piece of information (hence the term “flag”). No value is allowed. A prime example is *uk-bus:DirectorSigningReport*, which associates the signing of the Accounts with a particular Director via a specific member of the *EntityOfficers* dimension in the context of the item.

Version 2 of the transformation registry includes a new transformation, *ix:nocontent* which allows for a selection of arbitrary text to be associated with an empty XBRL concept. Until such a time as version 2 of the transformation registration is supported the *ix:exclude* element can be used to simulate this behaviour as follows;

If we assume that the name “Joe Bloggs” appears on the face of the document as the Director signing the Accounts then the following “pattern” will allow the Inline XBRL mark-up of this text with a “flag” item such as *uk-bus:DirectorSigningReport*:

```
<ix:nonNumeric name="uk-bus:DirectorSigningReport"
contextRef="CY"><ix:exclude>Joe Bloggs</ix:exclude></ix:nonNumeric>
```

#### 4.10 Use of typed dimensions (FRS 101, FRS 102 and EU IFRS Taxonomies)

The Irish extension taxonomies for FRS 101, FRS 102 and EU IFRS are based upon the Financial Reporting Council (FRC) taxonomies corresponding to those accounting standards as they apply in the UK. Typed dimensions replace the use of the tuples mechanism in the new taxonomies and should be used for supporting groupings and analysis items in the same way that tuples were.

The FRC have published documents on the design of their taxonomies, together with a tagging guide. Links to those documents are below.

[https://www.frc.org.uk/documents/7765/Accounts\\_Taxonomies\\_Design\\_2025.pdf](https://www.frc.org.uk/documents/7765/Accounts_Taxonomies_Design_2025.pdf)

Information on typed dimensions can be found in Section 5, while information on the creation of analysis items is in Section 6.2, with information on how typed dimensions replace tuples as the means of supporting groupings in Section 7.

[https://www.frc.org.uk/getattachment/8fd53609-311d-431e-800d-805d4252e190/FRC-XBRL-Tagging-Guide-%E2%80%93-FRC-Taxonomies\\_October-2021.pdf](https://www.frc.org.uk/getattachment/8fd53609-311d-431e-800d-805d4252e190/FRC-XBRL-Tagging-Guide-%E2%80%93-FRC-Taxonomies_October-2021.pdf)

Information on typed dimensions can be found in Section 3.6.5, while information on how typed dimensions replace tuples as the means of supporting groupings is in Section 3.7, with information on the creation of analysis items in Section 3.8.

## 5 FURTHER INFORMATION

Further information including technical notes, an error message guide and details of test facilities can be found on the Revenue.ie website at the following location:

<https://www.revenue.ie/en/companies-and-charities/corporation-tax-for-companies/submitting-financial-statements/technical-information.aspx>

If you have any queries in relation to this guide, please contact us at [ixbri@Revenue.ie](mailto:ixbri@Revenue.ie).