

17. Human Readable Format and XBRL

Being able to read the information in an XBRL instance document is important as the preparer needs to review that document and someone consuming the information also needs to be able to read that document. Financial information is complex and formatting helps keep the information in context making it more understandable.

17.1. Difference Between Formatting, Rendering, Styling and Presentation

It is worth clarifying the definitions of formatting, rendering, styling, and presentation for the purposes of this document as different people have different definitions of these concepts.

17.1.1. Formatting

Formatting is the process of assigning information to XBRL in order to later render the information in some desired format. For example, XSL-FO is a formatting language. XSL-FO is designed so that it contains information to allow data to be rendered in a number of different formats.

17.1.2. Rendering

Rendering is the process of taking formatting information, such as XSL-FO, and using the formatting information to render a presentation. For example, XSL-FO can be rendered as PDF, HTML, text, Braille, audio, etc.

17.1.3. Styling

Styling is the process of assigning things like bold (or more properly called emphasis), italics, single underline, double underline, etc. to information. This also includes indentation,

Styling is transforming and formatting information.

Styling is the rendering of information into a form suitable for consumption by a target audience. Because the audience can change for a given set of information, we often need to apply different styling for that information to obtain dissimilar renderings to meet the needs of each audience. Perhaps some information needs to be rearranged to make more sense for the reader. Perhaps some information needs to be highlighted differently to bring focus to key content.

17.1.4. Presentation

Presentation is the process of presenting data in a certain manner. For example, the data value "0" (or zero) can be presented in a number of formats within financial information, according to the preferences of the preparers of this information, such as:

- 0
- -
- -0-
- (blank)

Further, there is a difference between "0" and nil, as defined by XBRL. Nil means no data is present, and is NOT the same as "0".

17.2. Introduction

First off, it is important to clarify the issue which is being discussed in this document as many people have use the same term in different ways. This document addresses "formatting" as defined in the terms above.

Others refer to this issue as "rendering" or "presentation" or "styling", but we will use these terms as defined above.

To frame this discussion, formatting information (which includes styling information) is used to render information into a number of formats, such as: output formats (Word, Excel, PDF, HTML, text, whatever) or one or more input formats (Word, Excel, HTML, PDF, XForms, or whatever).

So, while the end goal may be a human readable rendering of information, the form of that rendering may be different for different consumers or producers. In addition the styling desired could be different, whether for example, emphasis is bold on paper, or a "louder voice" in an aural system used by the hearing impaired or blind to review a business report, such as a financial statement.

Finally, many examples of formatting issues are discussed in this document, and many of those relate to financial reporting. But this document addresses the general case of business reporting, of which one use case is financial reporting. It is the case that external financial reporting may be the toughest use case to address. If so, if that use case is addressed, all other business reporting use cases will be addressed. This may or may not be true; as it may or may not be true that there needs to be separate ways to express formatting information for external financial reporting and business reporting in general.

The purpose of this document is to (a) analyze these use cases and then (b) articulate a set of requirements from those use cases.

17.3. Overview

The primary purpose of XBRL is to exchange information. In the process of exchanging information, many times humans are involved in inputting, debugging, reviewing, and analyzing information. The ways humans interact with data and the ways computers interact with data are different. Accountants have come up with common ways of looking at information. While not "standards" per say, they are ways which work to visually create or consume financial information, which can be quite complex.

17.4. First Example

This first example will help to focus on the real issue, helping to throw out things which are not the issue and distract from understanding the issue. Consider the following two examples of information. The information shown is identical both examples.

Form 1: The following is information shown in one form:

Land and Buildings, Balance	1/1/2003	244508
Land and Buildings, Additions	2003	109659
Land and Buildings, Disposals	2003	-193
Land and Buildings, Translation difference	2003	12401
Land and Buildings. Balance	12/31/2003	<u><u>366375</u></u>

Furniture and Fixtures, Balance	1/1/2003	34457
Furniture and Fixtures, Additions	2003	0
Furniture and Fixtures, Disposals	2003	0
Furniture and Fixtures, Translation Difference	2003	0
Furniture and Fixtures, Balance	12/31/2003	<u>34457</u>
Other, Balance	1/1/2003	6702
Other, Additions	2003	7100
Other, Disposals	2003	-262
Other, Translation Difference	2003	-7487
Other, Balance	12/31/2003	<u>6053</u>
Total, Balance	1/1/2003	285667
Total, Additions	2003	116759
Total, Disposals	2003	-455
Total, Translation Difference	2003	4914
Total, Balance	12/31/2003	<u><u>406885</u></u>

Form 2: This is exactly the same information shown in a different form:

	As at 1/1/2003	Addition s 2003	Disposals 2003	Translatio n difference 2003	As at 12/31/200 3
Land and Buildings	244508	109659	-193	12401	366375
Furniture and Fixtures	34457	0	0	0	34457
Other	6702	7100	-262	-7487	6053
Total	<u>285667</u>	<u>116759</u>	<u>-455</u>	<u>4914</u>	<u>406885</u>

The information in both formats (the renderings are the same, both in this Word document) is identical. The first shows the information, but it is simply linear in nature, somewhat similar to how the information is contained in an XBRL instance document. Note the single and double underlines. It is quite easy for an XBRL processor to determine what needs to be underlined by using the calculation linkbase. Here, simply totals have a double underscore after them and a single underscore prior to the listing of "children" of the calculation. Any XBRL processor could easily achieve either.

But, how does an XBRL processor know to put certain calculation sets in columns and how to relate each column? A human can easily see this. But there is no way to express this information within a taxonomy. That is where XBRL falls short, the information for determining which column to put the information in and how to relate the columns does not exist in XBRL.

Now, it is quite easy to provide this information, how to format instance document information and then generate formatted instance document information and renderings of that information in PDF, Word, Excel, HTML, RTF, text, or some other format. But, the first question is the following: do you want everyone using different ways to express the formatting information? Maybe it does not matter. But, it also may be the case that it does matter.

A second situation exists. Suppose you create a rendering in PDF for human consumption of the instance document information. Now you have "two versions of the truth." You have the XBRL instance document and you have a rendering. How do you know you have the correct matched set of XBRL instance and

rendering? How do you know you got all the information in the rendering from the instance? Accountants spend hours and hours in a process called "footing and cross-casting" a financial statement, "ticking and tying" it to the "source documents". Accountants put on their green eye shades and spend hours and hours making sure there are no errors, basically "debugging" a financial statement. Developers gave this manual testing process up long ago, using "conformance suites" and automated testing processes to help make sure software acts properly and has no bugs. Is it possible to do the same thing for financial statements; automate the process of determining that the rendering and the XBRL instance agree, and are both free from errors? Perhaps it is, and perhaps it is not. However, it would certainly be easier to build that software if there were one standard way of expressing the formatting information.

17.5. Second Example – Filings with the US SEC

Rendering (formatting) information is not the problem. It is 100% possible to render (format) XBRL. It is quite easy to render XBRL instance information, using XSLT, as: Word, Excel, PDF, HTML, text and other formats. This ease comes from the fact that XBRL is XML. You can do this, today, using XSLT tools which are on the market, they exist today. It is true, however, that these tools are not optimized for XBRL, and probably never will be. But they can, and do, work.

The problem is that there is no "public standard way" to express the rendering (formatting) information (rather than not being able to render/format). THAT is the issue. The question is, "what is the need for a public standard way to provide formatting information so that XBRL information can be rendered." There are other sub-issues such as how to deal with "prose", etc. But the primary issue is the "public standard way".

The situation can be exemplified by looking at the following scenario: Say the US SEC adopts XBRL for financial reporting, you will have thousands, millions of instance documents. The investing public needs to have no worse experience than they have now when they try and use this information. Now, using the information means they can go "look at the information".

In the future these users will be able to "look at the information" and in addition "grab the information automatically", if they so choose.

But the question is: How is (a) looking at the data achieved, and (b) how to you make sure what you look at and what you grab is the same information? Any thing beyond (a) and (b) is gravy on the meat and potatoes. But (a) and (b) ARE the meat and potatoes.

This CANNOT be done without having (a) every filer creating and providing "a way" which investors can view the information, (b) the SEC creating "a way" and mandating that all filers follow that way, or (c) the SEC adopting "a way" that someone else created and mandating that all filers follow that way of facilitating investors to view filed information (unless the way is a "defacto standard" so the US SEC don't even have to mandate it).

That is the spectrum of options. "The way" is to be determined by what people can (or cannot) agree on; that (does it include styling information such as "bold" or other emphasis, underlines, colored text, etc.) is a totally separate question from "the way". The biggest issue really is how to show contexts, information in tuples, and dimensional information. Does the information go "down and across" or "across and down" and in what column should the information appear.

17.6. *Is there a Need for a Public Standard for Formatting?*

Rendering (formatting) information is not the problem. It is 100% possible to render (format) XBRL. It is quite easy to render XBRL instance information, using XSLT, as: Word, Excel, PDF, HTML, text and other formats. This ease comes from the fact that XBRL is XML. You can do this, today, using XSLT tools which are on the market, they exist today. It is true, however, that these tools are not optimized for XBRL, and probably never will be. But they can, and do, work.

17.7. *Prose*

Another issue relating to formatting is prose. The following is an example of prose which might be included within a financial statement:

On 1 March 2002 the Group acquired a 100% interest in XYZ Limited which manufactures paints and is incorporated in Singapore. The consideration of LC 7,950 was settled in cash. The fair value of the net identifiable assets of the company at the date of acquisition was LC 5,145. This acquisition has not been recognised in the Group's 2001 financial statements. Goodwill arising on this acquisition of LC 2,805 will be amortised on a straight line basis over 10 years, reflecting the importance of this acquisition in expanding the Group's market coverage in the paints industry. The operating results and assets and liabilities of XYZ Limited will be consolidated with effect from 1 March 1999.

If you look at the above paragraph, you will note that you have pieces of discrete data intermingled within one paragraph of text. The issue is that, how would a user get to the one value the consideration amount of LC 7950? That value is buried in a sea of text. This is a micro example of the problem of having a financial statement be available only as a sea of text which is impossible to extract information from. Here the set of text is smaller, but the problem is the same.

Also, this information is intended to be consumed together. There are no real links between fact values which must be consumed together within an instance document to get the real picture of what is trying to get communicated from, say, a paragraph of text which a human can see goes together.

17.8. *Labels, Headings, Multiple Languages*

A financial statement is not all fact values, but includes other textual information and numeric information. Consider the following:

The following is a summary of Property, Plant and Equipment as of December 31,

	2003	2002
	€'000	€'000
Land	5,347	1,147
Buildings	244,508	366,375
Furniture and Fixtures	34,457	34,457
Computer Equipment	4,169	5,313
Other	6,702	6,149
Total	295,183	413,441

Note the following from the example above:

- Column headings for the periods 2003 and 2002.
- The text "The following is a summary of Property, Plant and Equipment as of December 31," which is not a part of the financial statement data, but rather a heading which describes a table of financial information contained in the instance document. Should this heading be included in the XBRL instance document?

The pallet of the participants of the financial reporting, and business reporting process in general, will determine what is, and is not, needed. The point here is only to bring this issue to light so these decisions are conscious, rather than unconscious.

17.9. Move from Paper Based to Electronic Financial Reporting

Fundamentally, clearly there will be a change from paper-based financial reporting and business reporting. That fact is rather clear. Business reporting, and in particular financial reporting cannot be disrupted in that process. The change will be an evolution, not a revolution, most likely.

Paper-based financial reporting evolved. Paper-based financial reporting has a number of characteristics due to the nature of the media, paper. For example, in financial reporting there is a distinction between "presentation" (which means of the face of the financial statements such as the balance sheet, income statement, cash flow statement, statement of equity) and "disclosure" (which means information is placed in the explanatory disclosures, or notes, of the financial statement. Yet, in XBRL there is no notion of "face of the financial statement". Does that notion need to be carried into XBRL, or can the distinction between "presentation" and "disclosure" be dropped as the media has made it obsolete?

The issue here is not what is right or wrong, but rather than the media has changed and there could be a tendency to bring paper-based "baggage" into XBRL, unnecessarily, and make XBRL work less well than it could have worked and the unnecessary baggage been dropped as a requirement.

17.10. What does an XBRL Financial Statement Consist of?

The purpose of this query is to get peoples ideas on what an "XBRL Financial Statement" really is, or should be. Maybe the question can even be stated better. But after reading this, I think the point in am trying to make is clear and a very valid point. Is XBRL enough? Or, do you still need a human rendering? Do you need a "trail" or is "traceability" required between the data and the human rendering?

Note that a set of files exists to show each of these options. (Please contact the author for the set of files.) Although these files are technical, and this discussion may be technical in that it explains using very tangible examples (the physical files); this is really about a decision, not about the technology as to how to GET to the end result. If the Spec group can come up with other ways, cool. I speculate that these XML-based approaches will likely be the way we get to what we desire.

But this discussion is more about, "What do we desire?" Data only? Data and presentation? Data, presentation, and an "trail" (traceability) between the XBRL instance and rendering? Something else?

These are the options to “XBRL Financial Statement” that I can see, perhaps others can see additional options. These options are articulated below, and physically expressed in the sample files provided.

17.10.1. Options

So, options are set forth below. It is believed that this set of options covers the spectrum of possibilities. No option is deemed better or worse than the other, this is all about simply articulating the options.

17.10.1.1. Option 1 – XBRL Instance Only

Is the XBRL financial statement the instance document alone? (BasicCalculation-instance.xml)

This is just “the data” in an XML format. No taxonomy is provided. But without the taxonomy (the meta data which explains the data), the instance is pretty much useless. Also, there is no rendering for human consumption.

So, under this option you get an instance document filled with XML, no taxonomy to describe the metadata in the XBRL instance document. There is no “human readable” form.

17.10.1.2. Option 2 – XBRL Instance and Taxonomy (DTS)

Is the XBRL instance document and its set of taxonomies enough? DTS is defined as the set of all taxonomies required by the instance document (BasicCalculation-instance.xml, BasicCalculation.xsd, and all the other taxonomy files)

Under this option we have data and meta data. Maybe sometimes this will be sufficient. But you still have no human readable rendering, which is very important when a human reads the data in an instance document. Is it the case that the instance document data can be deemed as separable from the formatting? Formatting provides visual cues as how to read the information. The financial information, which is highly technical in nature, is virtually impossible for a human “consume” (understand, derive meaning from) without this organization provided by the formatting/styling. This is NOT about “bold” or italics. It is about, say, organizing a movement analysis how an account EXPECTS to see and work with a movement analysis.

17.10.1.3. Option 3 – XBRL Instance, DTS, Style Sheet

Is the XBRL financial statement the DTS (instance plus taxonomies) PLUS the style sheet which renders the instance document into a human readable form? (Instance, taxonomy, + BasicCalculation-ToFo.xsl)

With this option you get the data, metadata, formatting information, but NOT the actual rendered human readable format. The style sheet actually gives you a “trail” (tractability) between the instance document data and the human readable rendering. It is not the world’s best trail if you don’t understand XPath/XSLT, but that trail can be made more human usable so you can “trace” or otherwise “audit” the path between the actual data and the human rendering.

So, here you have data, human readable, and a PATH from one to the other. This option does not include the ACTUAL RENDERING, but rather all the information to create a rendering when you need one.

17.10.1.4. Option 4 – XBRL Instance, DTS, Style Sheet, FO Output

Is the XBRL financial statement the DTS, PLUS the style sheet, PLUS, say, the XSL-FO which contains formatting information which can be consumed by an FOP (formatting objects processor)? (Instance, taxonomy, style sheet, + BasicCalculation.fo)

The formatting information (which is created by running the XSLT against the instance document(s) and taxonomies) contains all the information, but there are NOW NO LINKS BACK to the original instance document or XSLT which generated the FO input.

But what this FO output does is allows the user to RENDER the output into whatever format they desire, be it PDF, HTML, text, RTF, Word, etc.

So you get data, metadata, the path to the rendering, and the actual rendering; which you can discard and create other renderings, if you don't like the rendering you were given because you have the path.

17.10.1.5. Option 5 – XBRL Instance, DTS, Style Sheet, Actual Rendering

Is the XBRL financial statement the DTS, PLUS the style sheet, PLUS the FO output, PLUS the actual rendering, say in PDF? (Instance, taxonomy, style sheet, FO output, + PDF file)

So, here we for the first time have something consumable by humans. The PDF can be "consumed" by a human, visually. But, there is no link between the rendering of the data and the data itself in the PDF file, or in the XBRL file. You need the intermediate style sheet to get that trail/traceability. This could be a problem. In addition, the rendering could be in English, but the user desires Japanese labels.

So basically you are creating the end points (data and result) but no PATH to go from one end point to the other, unless you also provide the style sheet. The FO output can be regenerated, at any time, using the XBRL instance and the style sheet. The PDF can likewise be generated at any time by regenerating the FO output and feeding it to an FO processor.

17.10.1.6. Option 6 – Some Combination of the Above, Validation Reports

It is possible to have some combination of the above files.

Lastly, where does "validation" fit into all of this? See the file BasicCalculation-instance-calctrace.html. This shows that things add up. Should this play a role in what is included in the document set? Or, is the user expected to go validate the file (and therefore require an XBRL processor) to be sure it is valid each time they use the file? Or, is an instance validated only once, some sort of indication that the validation has occurred, meaning that you don't really need validation every time you use the file?

17.11. Style Sheets

There are many ways to convert XBRL or even XML into other formats; any application can be used. But, in the XML family, XSLT is the primary tool. These XSLT documents are commonly referred to as style sheets. A style sheet can be used to transform XBRL into a number of formats: PDF, HTML, Word, XSL-FO.

17.12. PDF

PDF is as pervasive as HTML. PDF and HTML have different features. The best feature of PDF is that it is used to create documents which have page breaks, footers, headers, and all the other stuff you would expect in a document.

17.13. HTML

HTML is intended to be used in a web browser. HTML has no page breaks. HTML will likely be used to as a rendering format for financial statements.

17.14. Word

The vast majority of all financial statements created in the world today are created using Microsoft Word. These financials may end up in many other formats, but they start out as Word documents.

17.15. XSL-FO

XSL-FO is actually a part of XSL. XSL has two parts: XSLT used to transform XML from one format to another and XSL-FO, or formatting objects (flow objects) which are used as input to a FOP, or formatting objects processor which will be discussed next.

XSL-FO is constructed to allow the rendering of documents – page breaks, footers, tables, etc. A primary output is PDF. Also, text and HTML outputs are commonly supported.

Generally, an XSLT style sheet is created which grabs information from an XBRL instance document and generates an XSL-FO XML document. The XSL-FO document is sent to a FOP for rendering as PDF, HTML, or text.

17.16. FOP

A FOP, or formatting objects processor, is an application which takes an XSL-FO document as input and generates a rendering, commonly PDF, but also HTML or text.

A FOP is a standard way of doing something in proprietary ways. There are freeware versions of FOPs available, and more robust versions which support more features of XSL-FO.

17.17. Summary

What form an XBRL financial statement will take is unclear, the XBRL International contortion has not specified any specific format, and there are not enough XBRL documents for there to be an ad hoc format which everyone seems to be using.

This will be determined in the future, probably based on what meets the needs of the creators and consumers of XBRL instance documents.