Embedding HTML markup in ONIX data elements

In ONIX – whether version 2.1 or 3.0 – there are many common issues that arise when data providers embed HTML within the various textual data elements. Data providers deliver HTML in a variety of different ways – some which match the standard, many which don't. And this means that for data recipients, the complexity of receiving so many unpredictable variations forces them to choose to just ignore all HTML – whether it matches the standard or not – or to treat each data file as unique (which adds unnecessary cost and time). This isn't good for either senders or recipients.

The primary reason to use HTML is to include multiple paragraphs, or italic or bold text, in an ONIX data element – a contributor biography or long description for example. Numbered or bullet point lists are also common requirements.

There are three standard ways in ONIX to include markup within a data element:

1 use XHTML without any other special treatment – this is the strongly preferred option
2 use HTML within the CDATA tag
3 use HTML, but escape all < characters to &lt; – this is the *least* preferred option

With option 1, you have to add the *textformat* attribute with the value 05. With options 2 and 3, you add *textformat* with the value 02. **Option 1 is strongly preferred in all cases.** If for some reason you cannot use XHTML, then 2 is preferred over 3 in ONIX 3.0, and in ONIX 2.1, 2 and 3 are equally preferred. **Pick an option, and use it consistently.**

For details of these methods, see the relevant sections in the ONIX *Specifications*. There are further explanations and examples in the *ONIX 3.0 Implementation and Best Practice Guide* – and even though this is a document for 3.0, the advice is equally applicable to ONIX 2.1.

Why is following the standard *exactly* so important? Right now, many data senders produce data that does not match the standard. And the huge variety of different errors means that many recipients have little option other than to ignore or strip out any and all markup. This results in long descriptions, author biographies or quotations from reviews appearing as monolithic and unattractive blocks of unformatted text when they are eventually presented on screen to potential readers. And many recipients strip the markup *even if you are one of the few senders that does it correctly*.

And why is the XHTML option preferred? XHTML is noting more than a slightly more rigorous version of HTML – the added requirements (at least for these purposes) are not difficult. All tags need to be lower case (so <p> instead of <P>) and all tags need to have matching end tags (every <p> must be matched by an </p>). With these simple requirements met, it eliminates the need for escaped < characters and CDATA tags. XHTML is still also correct HTML, so a recipient who 'requires HTML' is automatically able to deal with XHTML too. Using XHTML avoids some tricky technical problems that arise for recipients that post-process the ONIX using XSLT (a software technology for processing XML data). And best of all, validating the ONIX against the DTD or XSD schema also validates the structure of your XHTML markup, so you can be confident about the technical structure of your data.

Common errors

The first common error is simply omitting the necessary *textformat* attribute. **When embedding HTML, you must always include the attribute *textformat="02"***, as the default when you omit it is 'plain text'. If you're using XHTML, then you must use *textformat="05"*. If you omit *textformat* altogether, it means the data should be interpreted as plain text without markup – your markup may well be visible on the data recipient's web page as text.

Then there are common issues with the escaping of the HTML markup:

```
<Text textformat="02">&lt;p&gt;This is some text marked up with
HTML.&lt;/p&gt;</Text>

<Text textformat="02">&lt;p>This is some text marked up with
HTML.&lt;/p></Text>
```

Only the second version is correct according to the standard. **The rule is – if you switch [only] &lt; back to a < character, the data should be correct HTML**. In theory, if the data above is processed using an XML-aware processor, the two versions would be considered identical, but in practice, many data recipients do not use end-to-end XML-aware processing. Do not rely on &gt; in HTML tags being replaced by >.

```
<Text textformat="02">&lt;p>This is some text that mentions Marks &amp;amp;
Spencer.&lt;/p></Text>

<Text textformat="02">&lt;p>This is some text that mentions Marks &amp;
Spencer.&lt;/p></Text>
```

Again, only the second version is correct. **Don't 'double-escape' any characters.**

Because of the common issues with 'double-escaping', the CDATA method is strongly preferred over the escaping method in ONIX 3.0:

```
<Text textformat="02"><![CDATA[<p>This is some text that mentions Marks
&amp; Spencer.</p>]]></Text>
```

With CDATA, the data content should be valid HTML without any changes.

Note that CDATA should not be used on any data *except* HTML – don't use it unnecessarily on XHTML, or on ordinary plain text without markup. This is strongly discouraged:

```
<Text textformat="06"><![CDATA[This is some plain text that mentions Marks
&amp; Spencer.]]></Text>
```

Ideally, the whole text should be enclosed in <p> or maybe <ul> tags (or similar 'block level' HTML or XHTML tags):

```
 <Text textformat="02">This is some &lt;em>text&lt;/em> marked up with HTML.
&lt;p>Here is a second paragraph.&lt;/p></Text>

<Text textformat="02">&lt;p>This is some &lt;em>text&lt;/em> marked up with
HTML.&lt;/p>&lt;p>Here is a second paragraph.&lt;/p></Text>
```

Once again, only the second version is really correct. **There should be block-level tags enclosing the whole of the text – no free-floating text that is outside of any tags at all.**

And finally, with both 2.1 and 3.0, **using XHTML instead of HTML is the best method of all**.

```
<Text textformat="05"><p>This is some text that mentions Marks &amp;
Spencer.</p></Text>
```

What's the difference with XHTML? In many cases, not much. XHTML uses *textformat="05"* instead of *"02"*, and you must ensure that all the markup tags are lower case, properly matched and nested. In HTML, the </p> and </li> tags are actually optional, but in XHTML, they are mandatory, and XHTML tags must always be lower case whereas in HTML upper case tags can be acceptable).

What markup tags are available?

When ONIX initially introduced HTML markup of textual data (way back in mid-2000), there were no documented limitations on the range of HTML tags that could be included. When XHTML was introduced (in 2003) some limitations were documented. But these limitations were generous – perhaps too generous – and meant that most HTML tags other than those used for interactivity (things like forms and scripts) were technically okay. It meant that publishers could add complex HTML or XHTML tagging in an attempt to control the look and layout of their product descriptions on the retailer's web page, using HTML headings, tables, *style* attributes, even images and hypertext links. Understandably, this led many data recipients to simply reject *all* HTML and XHTML tags.

The technical limitations on XHTML remain, and are also still very generous. But in practice, there is a strong recommendation for data suppliers to limit themselves to using a very restricted range of simple HTML or XHTML tags – and the hope is that if data suppliers do this, data recipients will not simply reject all HTML out of hand.

It is strongly recommended that ONIX data suppliers use only the following tags:

<p> and <br /> – paragraphs, and line breaks within paragraphs
<i> and <em>, <b> and <strong> – italic and bold emphasis [1]
<cite> – for book titles
<ul>, <ol> and <li> – bulleted and numbered lists
<sub> and <sup> – sub- and superscript
<dl>, <dt> and <dd> – definition lists
<ruby>, <rb>, <rp> and <rt> – for simple glosses in Chinese, Japanese and other text

Of these, only <p>, <ul>, <ol> and <dl> are 'block-level'. Any attributes (e.g. the *style* attribute) should be avoided. And it should be emphasised that these recommendations apply to both HTML (using CDATA or escaping of the < character) and to XHTML.

There is complete list of XHTML and HTML tags – allowed and disallowed – within the *ONIX 3.0 Implementation and Best Practice Guide*, but the subset above is strongly recommended.

Which data elements allow markup?

There is only a small list of ONIX data elements in which HTML or XHTML markup is acceptable. For example, in ONIX 3.0, <BiographicalNote> can contain markup, but <ProductFormDescription> cannot. Here's the complete list of data elements where markup can be used:

- <AncillaryContentDescription>
- <AudienceDescription>
- <BiographicalNote>
- <BookClubAdoption>
- <CitationNote>
- <CopiesSold>
- <ConferenceTheme> (deprecated)
- <ContributorDescription>
- <ContributorStatement>
- <EditionStatement>
- <FeatureNote>
- <IllustrationsNote>
- <InitialPrintRun>
- <MarketPublishingStatusNote>
- <PrizeJury>
- <PromotionCampaign>
- <PromotionContact> (deprecated)
- <PublishingStatusNote>
- <ReissueDescription> (deprecated)
- <ReligiousTextFeatureDescription>
- <ReprintDetail>
- <SalesRestrictionNote>
- <Text>
- <TitleStatement>
- <WebsiteDescription>

In ONIX 2.1, HTML and XHTML should only be used in data elements which have an associated <TextFormat> element:

- <Annotation>
- <BiographicalNote>
- <DownloadCaption>
- <DownloadCopyrightNotice>
- <DownloadCredit>
- <DownloadTerms>
- <MainDescription>
- <PrizeJury>
- <ProductWebsiteDescription>
- <ReviewQuote>
- <Text>
- <TextWithDownload>
- <WebsiteDescription>

Q & A

Q. If the < character occurs in the text of a short or long description, and I also want to include HTML or XHTML markup, how should I do that?

Because the < character is used to mark the beginning of an tag, it can be confusing if you *also* want to include it as part of the text. Generally, if you want a < to appear in the text of a web page (not as part of the markup), you would use &lt;, and equally, if you want & to appear on a web page, you use &amp; instead.

But when you're embedding HTML in ONIX with the escaping method, < characters at the beginning of tags are changed to &lt;. The recipient will turn every &lt; back into a < character. And while that would be correct for < at the beginning of a tag, it would not be correct for a < that is meant to appear as part of the text.

The simplest way to deal with this is this: **if you're using the escaping method and you want a < to appear as part of your marked-up text, use &#60; instead of &lt;**

Alternatively, use the CDATA method instead. With CDATA, < at the beginning of a tag stays as <, and a < intended to appear as part of the text should be encoded as &lt; or &#60;.

And it is the same with XHTML: < at the beginning of a tag stays as <, and a < intended to appear as part of the text should be encoded as &lt; or &#60;.

Q. Can I include named character entities in text with markup in ONIX 3.0?

If the markup is HTML, this is pretty confusing, because named character entities – things like &hellip; for an ellipsis or &ndash; for a dash – are valid in HTML and XHTML. But they are *not* valid in ONIX 3.0. A strict answer to the question depends on which method you're using to embed the HTML:

`<Text textformat="02">&lt;p>This is some text &ndash; with a dash!&lt;/p></Text>` – not correct, because the named character entity isn't valid

`<Text textformat="02">&lt;p>This is some text &amp;ndash; with a dash!&lt;/p></Text>` – not correct, because the character entity is 'double-escaped'

`<Text textformat="02">&lt;p>This is some text &#8211; with a dash!&lt;/p></Text>` – correct, uses the numerical character reference for the dash

`<Text textformat="02">&lt;p>This is some text – with a dash!&lt;/p></Text>` – correct, uses the native character for the dash, in whatever character encoding the entire message uses

`<Text textformat="02"><![CDATA[<p>This is some text &ndash; with a dash!</p>]]></Text>` – this is correct – and will probably work – but may cause problems for data recipients which use XSLT post-processing

```
<Text textformat="02"><![CDATA[<p>This is some text &#8211; with a
dash!</p>]]></Text>
```
– correct

```
<Text textformat="02"><![CDATA[<p>This is some text – with a
dash!</p>]]></Text>
```
– correct

If you're embedding XHTML – with textformat="05", and without CDATA or escaping – then the answer is much simpler. Either the native character or the numerical reference are fine, but the named entity (&ndash;) is not.

Q. My ONIX uses Unicode and UTF-8, as the standard recommends, but the data recipients that I send data to say their web pages are still limited to Latin-1 or Windows-1252. I want to include a handful of Cyrillic characters in the text, and my system can cope perfectly well – but their system apparently can't. What should I do?

Well, that sounds complicated. But it can be done. First check whether they can cope with the character set conversion internally, either as they import the ONIX into their internal system or as they sent it out to their web content management system. It may just be okay.

If not, their online store web pages will still work if you can pass them either named character entities or numerical references for the Cyrillic text. Even if they say 'our web pages are limited to Latin-1', characters outside this small set will still work okay when they eventually get displayed within a web browser. So if you want a Cyrillic character Д, you can do it with &#1044; (or &#x0414; in hexadecimal). And if you're using ONIX 2.1, the named character entity &Dcy; will also work.

Notes

1. note there is a difference between <i> and <em> in HTML. While *typographically* then have the same effect – by default, they italicize the enclosed text – *semantically* they are different. The <em> tag italicizes for emphasis. The <i> tag italicizes for another reason, without adding emphasis – for example because a word is in another language, to indicate a technical term or taxonomic designation, or perhaps to indicate inner thought rather external action. The <cite> tag also indicates italicization, but is semantically distinct – it should be used to italicize titles (*eg* titles of books). Similarly, there is a semantic distinction between <b> and <strong>.