

ONIX for Books

Product Information Message

Application Note: Another Twelve ONIX Problems – common issues faced by ONIX senders

Senders of ONIX data face many problems – the ONIX message format is relatively complex because it aims to solve a relatively complex problem, and the number of potential data recipients is large enough to cause issues. This is particularly the case in markets where there is no strong central data aggregation service, because recipients process the same ONIX in different ways, or demand senders provide some needless variation in the data files – this often gets termed the ‘flavours of ONIX’ problem. Difficulties can be exacerbated by the variety of ways ONIX is used. As a result, senders are often forced to prepare unique files for particular recipients, making allowances for each recipient’s idiosyncratic requirements but at the same time adding largely unnecessary cost and complexity to the data supply chain. ONIX is intended to facilitate automated exchange of large volumes of rich metadata mostly without resorting to bespoke data files, and any departure from the standard comes at a cost.

Having said that, there’s no doubt that ONIX recipients face tougher challenges than senders, and this Application Note should be read alongside the original *Twelve ONIX problems*, a list of issues caused by senders.

All the following issues were nominated by major ONIX data senders, based on the real-world ONIX they send and the way they see recipients processing the data. Several senders contributed to the list – and their contributions demonstrate that these issues are extremely common and apply to even the biggest data recipients. The list originally appeared on the onix@groups.io mailing list (join the list by sending a blank message to onix+subscribe@groups.io).

ONIX recipients – does your ONIX ingestion, processing and display of data provided to you exhibit any of these issues?

12. Recipient requirements to omit certain data fields

It’s decreasingly common for data recipients to specify that they cannot deal with particular ONIX data fields, and that those unusable fields and composites must be removed before data is submitted. However, this continues to be an issue with some ONIX recipients.

Such requirements almost inevitably demand preparation of unique ONIX messages for that recipient, increasing the cost to serve and support that recipient.

The bigger danger, of course, is that limitations in processing at just one or two data recipients encourage a ‘lowest common denominator ONIX’ that lacks richness and expresses only the most basic metadata and commercial information, while failing to capture the complexity of real-world business arrangements. To avoid this, it’s important that data recipients engineer systems that *ignore* unwanted data rather than demanding its removal.

But not all data elements are ignorable...

There are certain data elements and composites that modify the meaning of other data – for example the way that <SalesRestrictions> modify the meaning of <SalesRights>, the way that data enclosed in <MarketPublishingDetail> (in Block 6) overrides key data provided in Block 4, or the way that a sales embargo date constrains retail sales. These cannot simply be ignored by data recipients because to do so is likely to lead to breaches of the conditions of sale.

11. Ignoring key data that is provided correctly

– for example, ignoring a Promotional headline provided in Block 2. A Promotional headline – something like ‘The most explosive thriller you’ll read this year!’ should be provided like this below, and displayed prominently to the consumer, either as a headline above a short or long description, or close to the title.

```
<TextContent>
  <TextType>10</TextType>           <!-- promotional headline -->
  <ContentAudience>00</ContentAudience>
  <Text>The most explosive thriller you'll read this year!</Text>
</TextContent>
<TextContent>
  <TextType>02</TextType>           <!-- short description -->
  <ContentAudience>00</ContentAudience>
  <Text textformat="05"><p>In Jim Donaldson's latest novel, lawyer Bob Searle
    faces his toughest challenge yet in a quest to prove an international
    business tycoon is attempting a coup in oil-rich Syldavia.</p></Text>
</TextContent>
```

This could be displayed within an online store something like this:



Searle in Syldavia

By Jim Donaldson

£7.95

Paperback

The most explosive thriller you'll read this year!

In Jim Donaldson's latest novel, lawyer Bob Searle faces his toughest challenge yet in a quest to prove an international business tycoon is attempting a coup in oil-rich Syldavia.

Data recipients' inability to use the Promotional headline in this way limits publishers' delivery of an important part of their marketing message, and encourages data suppliers to put this promotional text in an inappropriate field like the subtitle – in turn rendering *real* subtitles much less usable and causing extra work where inappropriate marketing text needs to be removed.

The point here – and the reason it's different from problem 12 – is that the recipient isn't ignoring the entire <TextContent> composite, or the <Text> field. Recipients are making use of descriptions, reviews and other text that's provided in the same way as the promotional headline. They are ignoring the content of <Text> when the value of <TextType> indicates a specific type of text (value 10).

Another common example of this issue is ignoring the <ContentAudience> data element that is used to distinguish marketing material aimed at the consumer from marketing material aimed at the book trade itself:

```
<TextContent>
  <TextType>02</TextType>           <!-- short description -->
  <ContentAudience>00</ContentAudience>   <!-- unrestricted -->
  <Text>Description that should appear on B2C websites...</Text>
</TextContent>
<TextContent>
  <TextType>02</TextType>           <!-- short description -->
  <ContentAudience>02</ContentAudience>   <!-- trade-only messaging -->
  <Text>Description that should only appear in B2B contexts...</Text>
</TextContent>
```

Recipients' common assumption that there will be only a *single* short description means that publishers cannot use <ContentAudience> with confidence. The risk is that the 'trade-only' B2B text appears unpredictably and inappropriately in a consumer-facing context. Yet many publishers do need to deliver different messaging to consumers, to trade buyers, to librarians and so on.

Two lesser-known fields that are often ignored by recipients are the Trade and Public announcement dates:

```
<PublishingDate>
  <PublishingDateRole>10</PublishingDateRole>  <!-- trade announcement date -->
  <Date>20220106</Date>
</PublishingDate>
```

These dates express embargoes on the redistribution of metadata to other trade organizations and to the public. They allow a publisher to provide metadata early to a specific supply chain partner – a data aggregator for example – without it being passed on until the embargo expires. If a recipient doesn't implement these controls, a publisher may be forced to delay data delivery to that recipient (while delivering the data earlier to partners who do implement the embargoes).

10. Ignoring repeats of common composites

The example above with repeats of the short description differentiated by their Content audience is an instance of a broader issue: many recipients make assumptions about data elements that are repeatable – they accept only *one* repeat of several, and of course *which* one they accept varies...

```
<TextContent>
  <TextType>06</TextType>                                <!-- review -->
  <ContentAudience>00</ContentAudience>
  <Text>'snippet of first review'</Text>
  <TextAuthor>Nia Esposito</TextAuthor>
  <SourceTitle>New York Times</SourceTitle>
</TextContent>
<TextContent>
  <TextType>06</TextType>                                <!-- review #2 -->
  <ContentAudience>00</ContentAudience>
  <Text>'snippet of second review'</Text>
  <TextAuthor>Julie Wilkinson</TextAuthor>
  <SourceTitle>Pocklington Weekly Advertiser</SourceTitle>
</TextContent>
```

Ideally, recipients would be able to use both of the review snippets above. But some recipients may select only the first, and others only the last review – though of course the first in the example above is likely to be commercially much more valuable to all parties. Multiple reviews should be expected, in separate repeats of <TextContent>. Unfortunately, with multiple reviews, the order cannot be made explicit, though this may be added in a future revision of ONIX. At present, data recipients should endeavour to prioritise reviews in the order they occur in the XML. Each review will potentially include the reviewer's name and the source publication in the <TextAuthor> and <SourceTitle> elements, and there could even be a Text source description to add credibility to the review:

```
<TextAuthor>Julie Wilkinson</TextAuthor>
<TextSourceDescription textformat="05">Author of <cite>Goat yoga for health and
  vitality</cite></TextSourceDescription>
<SourceTitle>Pocklington Weekly Advertiser</SourceTitle>
```

Recipients should never insist that all reviews should be packed together in a single instance of <TextContent>.

A lesser issue, but a frustrating one for academic publishers in particular, would be some small limit on the number of repeats of the <Contributor> composite that a recipient can accept. ONIX deliberately avoids placing limits on this repeatability. Even simple trade titles frequently have three, four or more contributors. Recipients who need to limit the number of contributors should consider using *at least* the first four (as denoted by <SequenceNumber>) and adding *et al* if there are more, rather than imposing a limit on all data senders. At least with contributors, there is an explicit <SequenceNumber>.

Allied to this is any recipient insistence that repeats occur *in a particular order* in the XML file – ‘you can have any number of product identifiers, but the GTIN must be the *first* identifier’. The <ProductIDType> tag should allow a recipient to select the GTIN they want, whether or not it’s the first identifier. The order of repeats of a particular composite should not be relied upon.

9. Reading only one <ProductSupply> (market), or only one <Price>

Issue 10 above referred to problems that arise when an ONIX data recipient is unable to use all the repeats of a particular composite – for example of <TextContent>, <Contributor> or whatever. But perhaps the most fundamental example where picking only one of a set of repeated composites leads to commercial misunderstandings is <ProductSupply>.

<ProductSupply> in Block 6 can be repeated to describe *multiple markets*, where for example a book might be for sale globally but is available in the Americas via one distributor, available in Europe and Africa via a different distributor and available in the Asia-Pacific region via a third. The book may have a consistent publication date in all three markets, or differing ‘local publication dates’ in each market (as in the example below). Almost certainly, pricing will vary between markets. Here’s an extended example:

```
<ProductSupply>                                <!-- first market - North America -->
  <Market>
    <Territory>
      <CountriesIncluded>CA US MX PR</CountriesIncluded>
    </Territory>
  </Market>
  <!-- no market-specific pub status, so inherits pub status from Block 4 -->
  <!-- no market-specific pub date, so inherits the global pub date from Block 4 -->
  <SupplyDetail>
    <!-- details of North American supplier and prices -->
  </SupplyDetail>
</ProductSupply>
<ProductSupply>                                <!-- second market - everywhere except North America -->
  <Market>
    <Territory>
      <RegionsIncluded>WORLD</RegionsIncluded>
      <CountriesExcluded>CA MX US PR</CountriesExcluded>
    </Territory>
  </Market>
  <MarketPublishingStatus>02</MarketPublishingStatus>    <!-- forthcoming -->
  <MarketDate>                                           <!-- in this market -->
    <MarketDateRole>01</MarketDateRole>                 <!-- *will be* published 7th Apr -->
    <Date>20220407</Date>                                <!-- 2022 in this market -->
  </MarketDate>
```

```

    <SupplyDetail>
      <!-- details of supplier and prices outside North America -->
    </SupplyDetail>
  </ProductSupply>

```

[In this example, the publication date (in Block 4) could be some time during February 2022. Then if the ONIX message were created in March 2022, the product would be published and active in the North American market, but remain ‘forthcoming’ in the second market (everywhere *except* North America).]

Where there are multiple <ProductSupply> composites, the contained <Market> and <Territory> composites should be used by the recipient to select the *right* <ProductSupply> ¹. For a reseller based in, for example, Singapore, a market where <Territory> does not contain SG is largely irrelevant.

Naïvely selecting only the first (or last) of several repeats of <ProductSupply> means the recipient remains unaware of the other markets. And the unwary recipient may well inadvertently ignore the market in which they themselves are located. Commercially this is likely to lead to resellers ordering from the wrong distributor, or to conclude (erroneously) that the product is not available to them.

Worst of all, of course, is when a recipient which can only use one <ProductSupply> composite demands the sender removes the other <ProductSupply> composites instead of selecting the one they need themselves. This creates the ‘flavours of ONIX’ problem and adds the cost to develop and deliver unique ONIX messages solely for that recipient.

The same can occur in other areas of the message too. “We can only accept one price per currency” remains an all-too-common response from recipients who demand data senders remove other prices that seem irrelevant *to them*. And yet multiple prices – for example repeats of the <Price> composite differentiated by date, for example – have been a part of ONIX from the outset and meet a commercial need to inform customers in advance of future price changes.

```

<Price>
  <PriceType>01</PriceType>
  <PriceAmount>9.95</PriceAmount>
  <CurrencyCode>USD</CurrencyCode>
  <PriceDate>
    <PriceDateRole>15</PriceDateRole>          <!-- Until -->
    <Date>20211231</Date>                       <!-- 31 Dec -->
  </PriceDate>
</Price>
<Price>
  <PriceType>01</PriceType>
  <PriceAmount>10.95</PriceAmount>
  <CurrencyCode>USD</CurrencyCode>
  <PriceDate>
    <PriceDateRole>14</PriceDateRole>          <!-- From -->
    <Date>20220101</Date>                       <!-- 1 Jan -->
  </PriceDate>
</Price>

```

¹ Note that if there is only one market and therefore only one <ProductSupply>, the contained <Market> and <Territory> composites are often omitted. In that sort of case, the market is the whole of the area with ‘for sale’ sales rights (<SalesRightsType> codes 01 and 02) in Block 4.

The example above shows a price valid until the end of the year and a new price beginning from the start of the new year.

8. Requirements to use non-standard codes, or worse, non-standard fields...

...or to add specific 'codewords' into particular fields, often in order to support some business process or sales model. One example might be to add a 'codeword' into an otherwise rarely-used textual field like <ProductDescription> or <PriceTypeDescription>. This is particularly frustrating for data senders if there is also a *standard* way to convey the same information, instead of mis-using a field. Here's an example:

```
<ProductForm>ED</ProductForm>                <!-- downloadable e-book -->
<ProductFormDetail>E107</ProductFormDetail>    <!-- PDF file format -->
<ProductFormDescription>Internet PDF</ProductFormDescription>
```

Now in one sense, 'Internet PDF' is meaningless – the downloadable file is either a PDF-based product or it's not. And if it *is* a PDF, then <ProductFormDetail> code E107 is the way to deliver this information in ONIX 3.0. In another sense, this 'Internet PDF' might be intended to indicate the PDF has been downsampled to a lower resolution or that it does not include crop marks that would be included in a 'Print PDF' intended for commercial printing.

But it's common to see this field used for some 'codeword' demanded by one data recipient as a signal to trigger some special business process. Each time this happens, it's another 'flavour of ONIX' that senders need to deal with, a bespoke ONIX file just for that particular recipient.

Of course, it might be that a recipient needs some new attribute in order to support their evolving business. Adding a 'codeword' to a rarely-used text field might be useful as a temporary workaround – though be wary of the legitimate uses of the chosen field. Using a codeword in a field devalues that field for other legitimate uses, just as adding a marketing message to a subtitle devalues that field for real subtitles.

But at the same time, EDItEUR provides opportunities to get new codes added to the 'official' ONIX codelists four times a year – and once a new code is added, the continued use of any workarounds should be avoided, because workarounds reduce the *interoperability* of ONIX. There are also uncommitted fields such as <SupplierOwnCode> that can be pressed into service without issue.

7. 'We support only the following codes'

Some data recipients provide 'data submission guidelines'. Transparency in how the data recipient treats data supplied is a good thing. But it's not so good to read guidelines that state – however clearly – that for codelist X, the recipient supports only a small subset of the codes available. Inevitably this means that the data sender must map a multiplicity of codes used within their own internal system onto that small subset, and then the data sender must generate bespoke ONIX solely for that recipient. But every sender does that mapping independently, and those multiple mappings vary. Worse, each sender must maintain different mappings for different recipients.

In an ideal world, recipients would support all codelist values, and keep up to date with new issues of the codelists. But more realistically, *where a data recipient supports only a limited subset of codes in a particular list, the necessary mapping from the full list to the subset should always be applied by the recipient*. This way, fewer overall mappings need to be kept up to date, and each mapping is created and maintained by the party that has direct knowledge of its own codelist subset so the recipient can be more confident of the mapped data's meaning. Best of all, data suppliers avoid the need to send yet another unique version of their ONIX data to that recipient.

6. Not using ‘statement’ fields like contributor statement and ‘rolling their own’ display fields algorithmically

In ONIX 3.0, there are a number of ‘statement’ fields which can be used by a data sender to indicate a preferred wording for something that might be tricky to construct from the content of individual fields.

Perhaps the best example is <ContributorStatement>. In cases where there is a single contributor, or multiple but unrelated contributors, each with a distinct role, it’s fairly simple to ‘construct’ a usable contributor phrase to display. If the contributor name is James Green and the role code is A01 (‘By’, or author), then displaying a phrase like ‘Written by James Green’ or simply ‘By James Green’ is fine. But when joint contributors are closely related or married, when a contributor has a dual role, or in more complex multi-contributor cases, it’s common for a data recipient to construct a wording algorithmically that turns out to be awkward to read – and it’s equally common for data senders to prefer a particular wording for that display phrase.

So for example, the display phrase ‘By Jon and Olivia Osbourne’ may be preferable to ‘By Jon Osbourne and Olivia Osbourne’, and ‘Written and illustrated by Fiona Chorlton’ reads less awkwardly than ‘Written by Fiona Chorlton, illustrated by Fiona Chorlton’. These preferred wordings can be supplied in <ContributorStatement>.

While each contributor must have a separate <Contributor> composite, and must have their own distinct name in the various <PersonName>, <PersonNameInverted> and structured name fields (<KeyNames> and so on), the Contributor statement allows this preferred wording to be included – additional to the main name metadata. And if it’s used for the intended display purposes only, it removes the need for a data recipient to ‘construct’ a statement for display to consumers that may appear awkward or stilted.

Of course, statement fields are not always provided, so the recipient has to construct a text string for display by concatenating roles and names in the normal way. And in most cases, an automatic, algorithmically-constructed string will be just fine. But where it is provided, the display string preferred by the data sender could – and maybe should – be used instead of a constructed string.

The most commonly-used instances of ‘statement’ fields in ONIX 3.0 are probably <ContributorStatement>, <TitleStatement> and <EditionStatement>.

The Title statement is often overlooked, but sometimes it’s valuable because it provides an opportunity to include some simple HTML markup in a title. The main title fields like <TitlePrefix> and <TitleWithoutPrefix> must be plain text – but for a work of literary criticism like ‘James Green and his critics’ which carries a subtitle ‘Post-modernist themes in “The poetry of landscape”’ (where ‘The poetry of landscape’ is a work by James Green, the <TitleStatement> could be:

```
<TitleStatement textformat="05">James Green and his critics: post-modernist themes  
in <cite>The poetry of landscape</cite></TitleStatement>
```

and the statement could be displayed as ‘James Green and his critics: post-modernist themes in *The poetry of landscape*’.

Note that these ‘statement fields’ can never *replace* the main title (or edition, or contributor name) fields – they always represent *additional* information that should be used for display purposes only.

5. Failure to process updates, or a tendency to stop updates at some point in the lifecycle

A common complaint heard from publishers sending ONIX to many recipients is ‘we sent an updated ONIX file, but the updates never appeared in their customer-facing system’. This tends to make data senders reluctant to send data early, because of the likelihood that it will need to be updated. It generates many calls for manual updates on online stores, and it can impact the author–publisher relationship when outdated metadata *continues* to be displayed (particularly to the public).

ONIX data is intended to be updateable at any point in the product lifecycle, to amend some metadata fields, to delete existing fields or add new metadata. Any full Product record received should update (*ie* replace) all the metadata previously received for that product. (NB a partial Product record – a ‘block update’– replaces only part of the previously-received metadata. EDItEUR has an application note on block updates available ²).

If the data recipient’s business processes mean they cannot accept updates automatically, or cannot accept them in certain circumstances, or cannot process them within a commercially reasonable time – no more than a few days – then this should be made clear in advance to data senders. This may potentially make data senders more reluctant to provide metadata at all (whether ONIX or any other format).

There’s a specific and common issue linked to this when the data recipient is reluctant or unwilling to process an update that would imply deletion of some previous metadata – whether that’s the removal of a planned publication date when a product is postponed indefinitely, removal of a previously listed co-author, or removal of a particular supporting resource. But replacing something with nothing – an expected pub date with no date, for example, or three contributors with two – *is* sometimes the correct way to process an ONIX update.

Here for example is part of the ONIX record for a planned publication date

```
<PublishingStatus>02</PublishingStatus>           <!-- forthcoming -->
<PublishingDate>
  <PublishingDateRole>01</PublishingDateRole>      <!-- publication date -->
  <Date>20220428</Date>
</PublishingDate>
```

This could be followed by an update:

```
<PublishingStatus>03</PublishingStatus>           <!-- postponed indefinitely -->
```

Note that there should be no publication date in the update when a product is postponed indefinitely (logically, there can’t be one...). Data recipients should then ensure they delete the old publication date at the same time as updating the Publishing status from ‘forthcoming’ to ‘postponed indefinitely’.

4. Creating author pages by matching names and not making use of publisher IDs or ISNIs

Many online retailers create ‘author pages’ – collating all the products for a particular contributor, so a potential purchaser can see not only the one book but all the books contributed to by a particular contributor. The collation of metadata for multiple books is almost invariably based on simple matching of the contributor’s name – and inevitably this results in author pages containing products from a mix of contributors who just happen to share the same name.

² <https://www.editeur.org/93/Release-3.0-Downloads/#How%20to>

Now sometimes this matching of names – putting all the books where the contributor name is the same on the same author page – is the only thing that a retailer can do. They don't really have any information to go on, other than the name. But in other cases...

```
<NameIdentifier>
  <NameIDType>16</NameIDType>                                <!-- ISNI -->
  <IDValue>0000000117803078</IDValue>
</NameIdentifier>
<PersonNameInverted>Taylor, Adrew</PersonNameInverted>
```

The solution? Name identifiers, whether standardized like ISNI or ORCID, or in the absence of standardized identifiers, proprietary identifiers like a publisher's own internal contributor ID. ISNI in particular provides a cross-publisher and cross-media solution that, when used consistently by publishers and resellers, could result in author pages that reliably list the offerings of one contributor – even products from other publishers – without a mixture of products from other contributors. There are 36 Andrew Taylors in the ISNI database at the time of writing (<https://isni.org/search>), each uniquely identifiable using their 16-digit ISNI. A retailer could create 17 different author pages, one for each distinct Andrew Taylor, and one combo page for all those other Andrew Taylors who don't have an ISNI.

And this could even extend beyond the world of books, to encompass music offerings by one of the Andrew Taylor, or links to TV or films Andrew Taylor was in, and so on. ISNI isn't limited to books, as its scope includes any creative sector. And as well as richer and more authoritative author pages, ISNIs are useful to libraries, for CIP and legal deposit processes. The one thing to remember is that an ISNI identifies a 'public identity', not a particular real-world person – you can have a different ISNI if you write under a pseudonym, or you could even share an ISNI with other people if you publish under a shared identity (see for example Lars Keppler, Michael Gregorio, Ellery Queen).

You can look up a particular name to find their ISNI via <https://isni.org/search>, and then add that ISNI to your ONIX data. But of course, not every contributor has an ISNI. The ISNI database is neither 100% complete nor 100% correct, but new ISNI assignments are being made all the time and corrections are processed where errors are spotted. If you do spot an error – where two ISNIs have been inadvertently assigned to one single public identity, for example, or where the ISNI database has a single ISNI for two distinct public identities, there's a reporting mechanism within the search page.

For a contributor who doesn't yet have an ISNI, you can apply via one of many ISNI registration agencies. They are analogous to the various ISBN registration agencies, though there's no country-based exclusivity as there is with ISBN. In principle at least, any reg agency can register ISNIs for anyone, though most are not yet ready to register identities on an ad hoc basis). Perhaps the easiest to use at the moment would be the British Library's registration portal at <https://isni.bl.uk>, but if you have a large number of contributors, contact one of the national ISNI agencies for advice (see <https://isni.org/page/isni-registration-agencies/>).

Bonus fact #1: the ISNI database also includes ISNIs for organisations. EDItEUR itself has an ISNI, [0000 0004 4914 9201](https://isni.org/0000000449149201), and so do many publishing companies, so they can be used as a <PublisherIdentifier> in ONIX.

Bonus fact #2: more-or-less the entire ISNI database is available free of charge – so you could download it and do your own lookups, rather than using the existing online search. There are a couple of caveats though. The free of charge data is up to a year out of date, so the latest ISNI assignments are not included, and the data format is not a simple table of rows and columns as you might expect – it's a 'linked data' format, either RDF/XML or JSON-LD. See <https://isni.org/page/linked-data/>.

3. Online retailers failing to display mandatory safety information

A significant amount of information is presented in ONIX's <ProductFormFeature> composite, ranging from responsible paper sourcing (FSC or PEFC credentials) and battery safety statements, to the accessibility information required by print-impaired readers (conformance with the EPUB Accessibility Specification, for example). Among this information can be various safety information and hazard warnings,

As an example:

```

<ProductForm>BK</ProductForm>           <!-- Novelty book -->
<ProductFormDetail>B213</ProductFormDetail> <!-- Book-as-toy -->
<ProductFormFeature>
  <ProductFormFeatureType>13</ProductFormFeatureType> <!-- EU Toy Safety
    statement -->
  <ProductFormFeatureValue>01</ProductFormFeatureValue> <!-- carries CE logo -->
</ProductFormFeature>
<ProductFormFeature>
  <ProductFormFeatureType>13</ProductFormFeatureType>
  <ProductFormFeatureValue>03</ProductFormFeatureValue> <!-- Carries '0-3' logo
    and warning -->
  <ProductFormFeatureDescription language="eng">Not suitable for children
    under 36 months, due to small parts</ProductFormFeatureDescription>
  <ProductFormFeatureDescription language="fre">Ne convient pas aux enfants
    de moins de 3 ans, contient de petites pièces susceptibles d'être
    ingérées</ProductFormFeatureDescription>
</ProductFormFeature>

```

In many countries, it's a legal requirement that when such warnings exist on the product itself, they must be presented on screen by online resellers, to ensure that the online purchaser receives the same warning as an in-person shopper. It's a particular issue for children's book publishers, who often include activity materials or toys that requires such warnings in their product range – and resellers should clearly ensure they're aware of the legal requirement in the countries they sell to.

Another often-forgotten requirement is the display of a 'required caption' – for example, use of an author photograph for promotional purposes may require use of a caption that clarifies the copyright of the image.

Bonus #3 – you'll see that the example above includes the text of the warning as 'multilingual metadata' in both French and English.

2. Lack of transparency, consistency and feedback

Almost without exception, data senders find it difficult to discover why their ONIX data – correctly constructed – is being interpreted incorrectly, why new data is ignored or updates fail, or why particular data fields they send are being displayed online in inappropriate ways, or not being displayed at all.

ONIX data recipient systems should not be 'black boxes', into which data senders are forced to fire a variety of experimental messages in order to gauge the results. The standard is intended to make such communication predictable, consistent and automated, in order to reduce costs across the metadata supply chain. And a big part of this is that ONIX messages that follow the standard should be interoperable – should be suitable for most or all recipients, without special per-recipient tailoring for special and non-standard requirements that add to the 'cost to serve' each client. Ideally, 'bespoke'

ONIX files created for individual recipients, should be limited to differences in the selection of Product records included, rather than incorporating non-standard modifications and workarounds.

Clearly there can be many reasons for a lack of consistency across different recipient systems, stemming from both technical limitations and human issues. Information about data recipients' systems needs to be made widely and more easily available, in comprehensive and up-to-date submission guidelines that outline not only what's required but also how it will be processed and displayed. And recipients' staff who are charged with liaison with data suppliers need to be fully conversant with both the technical capabilities and limitations of their own recipient system, and with ONIX itself, so they can provide reliable and consistent advice and feedback to data senders. Many senders note that a lack of consistency and transparency can be particularly frustrating with larger resellers, whose systems often appear to be global, or are described as being consistent across different store fronts and territories – but which are in fact a collection of disparate systems that work very differently in each country.

Inconsistent advice leads to a belief in a body of 'lore', ill-defined techniques that worked, once, in a specific circumstance, but which become widely believed and inappropriately applied for different types of books, in different territories, with different data recipients and well beyond their time. Sticking to the Standard is the only effective way to improve outcomes and reduce costs across the supply chain as a whole.

1. Still demanding ONIX 2.1

ONIX 3.0 is no longer 'new' – it's a stable and mature standard that's been around for more than a decade, and nearly all major ONIX users – senders and recipients – can support it.

ONIX recipients that *still* require ONIX 2.1 – and equally, data providers that can supply only 2.1 – are relying on a data format designed twenty years ago and which increasingly does not support today's business requirements. Continued reliance on the old version adds cost to the data supply chain for everyone else, and represents a business risk to both senders and recipients. For those 2.1 users who do not support 3.0: *it's long past the time to migrate.*

A final caveat

Of course some of these issues are more destructive than others. Indirectly encouraging data suppliers to misuse the <Subtitle> field is a negative for almost everyone (even for those other recipients who can use Promotional headline correctly, since they still get the promotional subtitles and may have to manually correct them). On the other hand, failure to list all contributors or displaying them poorly may only reflect badly on the particular data recipient. Others add costs across the data supply chain, particularly where recipients demand unique 'flavours of ONIX' and require data that is *unnecessary for them but vital to others* is removed.

For data senders, supporting a group of recipients with unique needs is expensive, error prone and should not be necessary.

This application note outlines a range of issues highlighted by ONIX metadata *senders*. A similar note is available at <https://www.editeur.org/93/Release-3.0-Downloads/#How%20to> that outlines issues seen by data *recipients*.

Graham Bell
EDItEUR
22nd December 2021

(reserve)

4. Failure to remove products from sale, or removing them from sale during a short period of temporary unavailability

A special case of the above arises at certain points in the product lifecycle, when a physical product is declared out of print, or when a product is deemed temporarily unavailable.

1. Out of print does not mean 'unavailable', and an 'out of print' publishing status may well be accompanied by an 'in stock' product availability. Resellers should keep a product on sale until it is both out of print *and* out of stock – and at that point it should be removed from sale.
2. In print but Out of stock does not mean 'unavailable' – it simply suggests that fulfilment time will be longer than usual. Depending on the reseller's policies on fulfilment, the product could remain on sale, accumulating orders that will be fulfilled once the product is available again.
3. Withdrawn from sale indicates a reseller should cease sales even if stock is available. Withdrawal might be for legal or technical reasons, or simply because the publisher has deemed the product 'not for sale'.