EDItEUR

ONIX for Books
Product Information Message

Application Note: Did you know…?

ONIX is a large and relatively complex message structure, and it is certainly true that no data sender or recipient makes use of all the options the format offers. The range of tags, and the range of codes from the various codelists, makes this practically impossible. For any one sender or data recipient, the range of functionality – the tags and codes used – should be chosen according to business need. Don't publish copies of the Bible? Then there are a range of tags you can safely ignore. Don't sell any children's books? Then as a retailer, you may choose to ignore any ONIX records that contain certain tags and codes.

But it's important to be aware of the range of possibilities, so recipients can ensure their file receiving and data ingest procedures are as robust as possible, and so that data senders can ensure no potential commercial benefits are missed. Both parties in a data exchange want to maximise product discoverability and sales.

This applicaton note highlighta a few of the lesser-known data an ONIX message can contain.

Did you know…

that ONIX includes two dates that can be used to control the onward dissemination of your metadata ?

It is implicit that if a publisher sends ONIX metadata to an organisation, it wants that organisation to make use of the data within the message. But some publishers wish to distribute data in advance to selected supply chain partners, under embargo, without necessarily making it immediately available to all and sundry. So how can you control whether that recipient organisation passes on the metadata to others?

- **Trade announcement date** – if you include this date, it indicates an embargo on any further distribution of the data. The recipient of the data should not disseminate the metadata at all, until that embargo has expired. The data indicates that it is solely for the recipient's internal use until the trade announcement date – at which time they can pass it on to their supply chain partners. Most often, this initial recipient is a metadata aggregator, or a distributor or wholesaler, and they can release the data to retailers on or after the trade announcment date
- **Public announcement date** – if you include this date, it indicates an embargo on dissemination of the data to the public – it should not be made available in a consumer-facing context by any recipient until the embargo expires.

Together, these dates can be used to send metadata to a data aggregator without that data being passed on immediately to everyone in the trade, or to send data to any business in the supply chain without that data being made immediately available to the public. The dates can be combined, so that the trade embargo expires before the public embargo, allowing a staged 'opening up' of the data. The example gives retailers around a week (from when they receive the data from a data aggregator) to get things ready before they go publc on their websites, and a further week before the sales embargo expires and the book is published:

```
<PublishingDate>
    <PublishingDateRole>10</PublishingDateRole>    <!-- trade annoucement date -->
    <Date>20190903</Date>                          <!-- aggregator releases -->
</PublishingDate>                                  <!-- data to retailer -->
<PublishingDate>
    <PublishingDateRole>09</PublishingDateRole>    <!-- public annoucement date -->
    <Date>20190910</Date>                          <!-- retailer releases data -->
</PublishingDate>                                  <!-- to general public -->
<PublishingDate>
    <PublishingDateRole>02</PublishingDateRole>    <!-- sales embargo date -->
    <Date>20190917</Date>                          <!-- actual sales begin -->
</PublishingDate>
<PublishingDate>
    <PublishingDateRole>01</PublishingDateRole>    <!-- publication date -->
    <Date>20190917</Date>
</PublishingDate>
```

Now these dates announcement dates have been a part of ONIX for many, many years – they were both present in ONIX 2.1 (*ie* in 2003). Despite this, not all recipients implement the embargoes correctly. Data senders who wish to make use of these metadata embargoes should definitely ensure in advance that their recipients can obey them reliably. And recipients should ensure their systems can keep data internal until the embargoes expire – or they risk receiving data later than their competitors as publishers are forced to provide bespoke ONIX that omits any sensitive data.

Did you know…

that ONIX includes a way of indicating which
of several contributors is pictured in an author photo ?

It's common to provide author photos as part of the marketing collateral for a book, but they are supplied in <SupportingResource>, not linked structurally to any particular <Contributor>. So if there are two or more contributors, how do you know which image applies to which contributor?

Actually there are two ways. First, any supplied image can have a caption that names the subject of the photo. Second, you can – perhaps more reliably – use an identifier (an ISNI or a proprietary contributor ID) to link the image with the contributor. Obviously, the identifier must match an ID that appears in one of the <Contributor> composites for the product. The example illustrates both options:

```
<SupportingResource>
    <ResourceContentType>04</ResourceContentType>      <!-- contributor picture -->
    <ContentAudience>00</ContentAudience>              <!-- for any audience -->
    <ResourceMode>03</ResourceMode>                    <!-- still image -->
    <ResourceFeature>
        <ResourceFeatureType>02<ResourceFeatureType>
        <FeatureNote>Steig Larsson</FeatureNote>       <!-- name in caption -->
    </ResourceFeature>
    <ResourceFeature>
        <ResourceFeatureType>05<ResourceFeatureType>
        <FeatureValue>0000000121337305</FeatureValue>  <!-- ISNI of contributor -->
    </ResourceFeature>
    <ResourceVersion>
        <ResourceForm>02</ResourceForm>                <!-- downloadable -->
        <ResourceVersionFeature>
            <ResourceVersionFeatureType>01</ResourceVersionFeatureType>
            <FeatureValue>D502</FeatureValue>          <!-- jpeg -->
        </ResourceVersionFeature>
        <ResourceLink>https://www.publisher.com/images/larsons.jpg</ResourceLink>
    </ResourceVersion>
</SupportingResource>
```

Of course this isn't necessary if there is only a single contributor, but it does no harm if you do add it.

The Resource feature composite can also carry a 'required credit' for an image. It is often a requirement of use to credit the photographer or copyright holder of a photo for example. As an ONIX sender, does your data include captions or identifiers when they are needed? As a recipient, do you display any required credits when they are supplied?

Did you know…

that ONIX includes a way to name the person a book is about ?

The <NameAsSubject> composite is most obviously useful for biographies, where the name of the subject of the book is a likely way for potential readers to search. But it can also be used like this:

```
<Subject>
    <SubjectSchemeIdentifier>93</SubjectSchemeIdentifier>            <!-- Thema -->
    <SubjectSchemeVersion>1.3</SubjectSchemeVersion>
    <SubjectCode>AGB</SubjectCode>      <!-- Individual artists, art monographs -->
</Subject>
<NameAsSubject>
    <NameIdentifier>
        <NameIDType>16</NameIDType>                                  <!-- ISNI -->
        <IDValue>000000012100026X</IDValue>
    </NameIdentifier>
    <PersonName>Paul Gauguin</PersonName>
</NameAsSubject>
```

This book is an art monograph, a study of a single artist and his paintings, who is named – and identified via an ISNI – using <NameAsSubject>.

In ONIX 3.0 [1], <NameAsSubject> has a structure very much like that of the <Contributor> composite – it reuses many of the same tags including <NameIdentifier>, <PersonName> (as above), and the elements of a structured name, dates and affiliations. You don't need to include a <NameIdentifier>, although these three examples all do.

<NameAsSubject> can also be used for corporate entities, and for fictional characters too. So a business book relating the foundation and history of a corporation might include something like this:

[1] in ONIX 2.1, <PersonAsSubject> has a similar function, but is more limited and can only be applied to real people.

```
<Subject>
    <SubjectSchemeIdentifier>93</SubjectSchemeIdentifier>            <!-- Thema -->
    <SubjectSchemeVersion>1.3</SubjectSchemeVersion>
    <SubjectCode>KJZ</SubjectCode>           <!-- History of specific companies -->
</Subject>
<NameAsSubject>
    <NameIdentifier>
        <NameIDType>16</NameIDType>      <!-- ISNIs can identify organisations! -->
        <IDValue>0000000109430267</IDValue>
    </NameIdentifier>
    <CorporateName>General Electric Company</CorporateName>
    <SubjectDate>
        <SubjectDateRole>50</SubjectDateRole>              <!-- date of ‘birth’ -->
        <Date dateformat="05">1892</Date>                       <!-- of company -->
    </SubjectDate>
</NameAsSubject>
```

And a fiction title that features a particular character might include this:

```
<NameAsSubject>
    <NameIdentifier>
        <NameIDType>16</NameIDType>     <!-- ISNIs for fictional characters too! -->
        <IDValue>0000000472411001</IDValue>
    </NameIdentifier>
    <KeyNames>Tintin</KeyNames>
</NameAsSubject>
```

Don’t overuse this last one – it’s really only intended for primary characters that span many books or even other media. It might be a good way for publishers to list the major characters in a superhero comicbook, or a protagonist who recurs over several books that are not a formal series.

Did you know…

that ONIX can tell you when a cover image is out of date ?

In the months leading up to publication, the publisher creates the cover of the book and distributes it to its supply chain partners. Wholesalers and retailers use this cover in catalogues and on websites, to encourage advance orders for the book. In ONIX, a link to the image would be carried within the <SupportingResource> composite, like this:

```
<ResourceLink>https://www.publisher.com/b2b/9780001234567-fc.jpg</ResourceLink>
```

and the idea is that a retailer can download the image from the given URL – though often, only a filename is given and some other process is used to get the actual cover image file to the retailer.

But once the retailer has the image, what happens if the publisher tweaks the cover a little, or changes it outright? How does the retailer discover that a new copy needs to be downloaded and used instead? This is what <ContentDate> is for:

```
<ContentDate>
    <ContentDateRole>17</ContentDateRole>
    <Date>20190620</Date>
</ContentDate>
```

This indicates that the cover was most recently updated on 20th June – so if the retailer's copy is older than this, the new version needs to be downloaded.

The Content date composite can also be used to carry other sorts of date – in particular, Content date role 14 means 'don't use the image before the given date', and role 15 means 'don't use it after the given date' – but the 'Cover last updated' date (code 17) is the most common by far [2]. ONIX recipients should look out for this and ensure there's a good process to obtain and use updated copies of cover images when they are available.

Did you know…

that ONIX can carry chapter-level metadata as well as information that applies to the whole product ?

Block 3 of ONIX 3.0 is dedicated to chapter-level metadata, and is probably of most use to publishers and retailers of academic and scholarly titles. Chapter level metadata [3] can include any of:

- an identifier for each chapter
- page (or time-based) ranges for each chapter
- title and any subtitle for each chapter
- contributors for each chapter
- subject codes and keywords for each chapter
- marketing collateral for each chapter (abstract or description, reviews, images etc)
- links to other products – for example if the chapter is available as a product in its own right

You'll notice that much of what can be said about a book can also be said about a chapter. And if the chapter is available as a product in its own right, then the full ONIX record for the chapter-sized product can mostly be included inside block 3 of the ONIX record for the book-sized product.

So you could imagine an ONIX record for a book forming the proceedings of a conference. The contributors at book level could be limited to the editors of the Proceedings, and the individual papers presented at the conference would be listed as individual chapters, with their individual authors, titles, abstracts and so on. Here's a short example, which would typically be packaged alongside detail of all the other content items (chapters) within <ContentDetail>:

```
<ContentItem>
    <LevelSequenceNumber>3.1</LevelSequenceNumber>
    <TextItem>
        <TextItemType>03</TextItemType>
        <PageRun>
            <FirstPageNumber>13</FirstPageNumber>
        </PageRun>
```

[2] in ONIX 2.1, the equivalent of the 'cover last updated' date is the <MediaFileDate> element, but there's no way in ONIX 2.1 to administer other media file dates such as embargoes like 'don't use before'.

[3] ONIX 2.1 can carry some but not all of the same data elements.

```
    </TextItem>

    <TitleDetail>
        <TitleType>01</TitleType>
        <TitleElement>
            <TitleElementLevel>04</TitleElementLevel>
            <PartNumber>1</PartNumber>
            <TitleText>Commodity Market Integration, 1500–2000</TitleText>
        </TitleElement>
    </TitleDetail>

    <Contributor>
        <SequenceNumber>1</SequenceNumber>
        <ContributorRole>A01</ContributorRole>
        <PersonNameInverted>Findlay, Ronald</PersonNameInverted>
    </Contributor>
    <Contributor>
        <SequenceNumber>2</SequenceNumber>
        <ContributorRole>A01</ContributorRole>
        <PersonNameInverted>O’Rourke, Kevin H.</PersonNameInverted>
    </Contributor>

    <Subject>
        <MainSubject/>
        <SubjectSchemeIdentifier>93</SubjectSchemeIdentifier>
        <SubjectCode>KCLT</SubjectCode>
    </Subject>

    <TextContent>
        <TextType>30</TextType>
        <ContentAudience>00</ContentAudience>
        <Text textformat="05"><p>This paper summarizes trends in international
commodity market integration during the second half of the second millennium.
International trade has increased steadily since the era of the European Voyages of
Discovery, and technology and politics have forced commodity markets to become more
integrated. But the trend has not been consistent – quite aside from major economic
shocks, politics has sometimes supported the effects of technological change, while
in others it has worked against them.</p></Text>
    </TextContent>

</ContentItem>
```

At it’s simplest, data like this could be used by the recipient to re-construct a table of contents (though there is a simpler way to send a basic TOC). More sophisticated use could present the purchaser with extremely detailed information about each chapter, leading not only better discoverability but also better conversion from interest into sales.

And it’s also possible to use <ContentItem> to provide timecodes for the start of each chapter in audio products, which audiobook retail platforms frequently request. In this type of data, <TimeRun> can replace <PageRun>.

Did you know…

that in ONIX 3.0, you can differentiate between prizes awarded to authors for their whole body of work, and prizes awarded for this particular book ?

The <Prize> composite in ONIX is intended to list details of any literary prizes won – whether it's the Nobel Prize for Literature, the prestigious Royal Society's Science Book Prize or a minor literary award. Such prizes provide a publicity boost, increasing discoverability and generating sales, so winning is worthwhile including in the book metadata.

The <Prize> composite usually looks something like this:

```
. . . </SupportingResource>
<Prize>
    <PrizeName>Prix Renaudot</PrizeName>
    <PrizeYear>2018</PrizeYear>
    <PrizeCode>01</PrizeCode>
</Prize>
```

This notes that the book won the Prix Renaudot last year – Prize code 01 means winner. Of course not every book can be a winner – so shortlisted titles or runners-up can also be listed too.

The above is for prizes won by the book itself. These appear in Block 2 (<CollateralDetail>) of ONIX 3.0 alongside other marketing-focused data. In contrast, generalised awards given to contributors for their whole body of work (such as the Nobel Prize) should not be listed in Block 2, nor should awards given to other books by the same author. But these can be listed – using exactly the same tag structure – within the relevant <Contributor> composite in Block 1 instead [4]:

```
. . . </ProfessionalAffiliation>
<Prize>
    <PrizeName>Rhône-Poulenc Prize for Science Books</PrizeName>
    <PrizeYear>1994</PrizeYear>
    <PrizeCountry>GB</PrizeCountry>
    <PrizeCode>01</PrizeCode>
    <PrizeStatement>Winner of the Rhône-Poulenc Prize 1994 for 'The Language of the
Genes'</PrizeStatement>
</Prize>
```

You can see it's also possible to add a <PrizeStatement>, which is a short text intended for display purposes and – perhaps for lesser-known prizes whose name is less clear, or as in this case, where the prize name might appear misleading – a country code.

Note that all this author's books could include this <Prize> listing – but it would be in a different place. The ONIX record for *Language of the Genes* itself would list it conventionally in Block 2. Books other than *Language of the Genes* would list it in <Contributor> (within Block 1).

[4] ONIX 2.1 is more limited – <Prize> is only suitable for prizes won by the book itself because it cannot be used within <Contributor>, and it lacks <PrizeStatement>.

Did you know…

that ONIX metadata can be provided in multiple languages simultaneously ?

As a standard intended for the global book trade, it’s natural that ONIX can handle metadata in many languages. Each of these four examples – in Chinese, English, Greek and Hebrew – just works, providing you have the text encoding right in your system:

```
<TitleText>三体</TitleText>

<TitlePrefix>The</TitlePrefix>
<TitleWithoutPrefix>three-body problem</TitleWithoutPrefix>

<TitlePrefix>Το</TitlePrefix>
<TitleWithoutPrefix>πρόβλημα των τριών σωμάτων</TitleWithoutPrefix>

<TitleText>בעיית שלושת הגופים</TitleText>
```

Now, these are likely to appear in four different Product records, or more likely in four different ONIX files from four different publishers, because each of the last three is a translation of the first book, an original Chinese novel. (Actually, the last one isn’t real, as the book does not appear to have been translated into Hebrew yet, but it illustrates that right-to-left text isn’t a problem.)

But sometimes you want to include multi-lingual text for the same product. For example, the book itself may be a bilingual edition in German and French, so you might perhaps want to provide bilingual metadata in both German and French. Of perhaps the book is in English, but you want metadata available in both English and in the language of the market the book is selling into (for example, selling English language material into China). To make this possible, many of the ‘textual’ data fields in ONIX 3.0 are repeatable, so you can just include something like this:

```
<BiographicalNote language="eng" textformat="05"><p><strong>Umberto Eco</strong>,
professor of semiotics at the University of Bologna, and author of <cite>The Name Of
The Rose</cite> and <cite>Foucault’s Pendulum</cite>, is one of the world’s
bestselling novelists.</p><p>As well as novels, he also writes children’s books and
academic works.</p></BiographicalNote>
<BiographicalNote language="ita" textformat="05"><p><strong>Umberto Eco</strong>,
professore di semiotica all’Università di Bologna e autore di <cite>Il nome della
rosa</cite> e <cite>Il pendolo di Foucault</cite>, è uno dei romanzieri più venduto
al mondo.</p><p>Così come romanzi, lui scrive anche libri per bambini e opere
accademici.</p></BiographicalNote>
```

Here, the same biography is provided in parallel, in both English and Italian. Both <BiographicalNote> elements must include the *language* attribute, whereas if you only have a single language it’s not at all necessary. Why? If there are two languages, the data recipient needs to know which is which, so the attribute helps them automate the choice. If there’s only one language, then the retailer doesn’t have a choice and can only use what’s there. You’ll also see the *textformat* attribute, which should be included whenever the biography includes any markup, as these two do ("05" means XHTML markup).

This repetition of ONIX tags for parallel multi-lingual data isn’t possible with every field, but it's allowed for more than 40 individual tags. See section X.14 in the *ONIX 3.0 Specification* for details. Perhaps obviously, it's not possible (nor is it necessary) for numbers, dates or anything that's drawn from a codelist.

Did you know…

that what we often call ‘country codes’ are not always codes for countries ?

ONIX Codelist 91 lists the available ‘country codes’ – codes like AT, HR, MX or TH used to represent Austria, Croatia, Mexico or Thailand. By and large, the country code represents the country – the whole of the country – but this isn’t true of all country codes.

France is among the more complex examples. In List 91, the code FR means what is sometimes called ‘metropolitan France’, that is, the parts of France that are clearly part of the European continent – ‘mainland’ France and Corsica. There are a dozen other parts of France spread around the world that are not part of metropolitan France and are not covered by the ONIX code FR, and yet these can also count as ‘France’ in various contexts.

So for example French Guiana, Martinique, Guadeloupe, Réunion and Mayotte are all ‘overseas départments’ of France, with much the same status as départments within Metropolitan France. Each of these has its own country code, GF, MQ, GP, RE and YT. And there are other parts of France, with different legal statuses, that should also not be forgotten – St Martin, Martinique, St Barthélemy, New Caledonia, St Pierre and Miquelon, Wallis and Fortuna, French Polynesia and the French Southern Territories (MF, MQ, BL, NC, PM, WF, PF, TF). In all that means up to 14 ‘country codes’ are needed to delimit France:

```
<Territory>
    <CountriesIncluded>FR GF MQ GP RE YT MF MQ BL NC PM WF PF TF</CountriesIncluded>
</Territory>
```

Of course this will vary, depending on what the publisher’s contract actually says and how that’s interpreted. Sometimes these various parts of overseas France (*'la France d’outre-mer*' is the antithesis of *'la France métropolitaine*') are included explicitly or by implication in ‘France’, and sometimes they are not. Your ONIX – for example in <SalesRights> – should reflect the full meaning of the publisher’s contract.

Corsica represents another good example. Unlike some other French islands, it is included automatically in the code FR – it does not have a country code of its own. So how can you say that sales specifically include or exclude Corsica? Well, it does have a ‘region code’ [5]. Regions in ONIX are parts of countries, and wherever in ONIX the <Territory> composite is used, extra regions can be added to or subtracted from the country. So to delimit metropolitan France without Corsica, you would use:

```
<Territory>
    <CountriesIncluded>FR</CountriesIncluded><!-- by default FR includes Corsica -->
    <RegionsExcluded>FR-H</RegionsExcluded><!-- then it is explicitly subtracted -->
</Territory>
```

[5] Not every country has regions available in ONIX List 91. If you need a region that does not have a code, contact EDItEUR. Lists are revised several times a year and a required region can be added.

And why might you need to use a construction like this? Because occasionally, parts of countries have different tax regimes, so a <Price> composite that includes tax might need to give separate details for that part of the country. An example would be the Canary Islands (region code ES-CN) – part of Spain (country code ES) and without a country code of its own, its currency is the same as the main part of Spain but very different taxes are applied (no VAT, but a kind of sales tax instead). So you might require two <Price> composites, one for ES excluding the region of ES-CN, and one just for ES-CN (this latter would use <RegionsIncluded>).

So what 'countries' actually comprise several country codes? France, Britain, United States, China are perhaps the most obvious, but Finland, Norway, Denmark, Netherlands, Australia and others also have similar dependencies, overseas or other territories with their own country codes.

ONIX's List 91 is 'based on' the ISO 3166-1 standard for country codes, though it's not exactly the same since codes are never removed from the EDItEUR list – they can be deprecated, meaning they should not be used in any new data, while remaining valid in old ONIX. The same is true of List 49 (regions), which is based on various parts of ISO 3166-2, and a range of other ONIX lists for currency, language, script and so on. Codes added to the ISO lists, and most changes to the ISO lists, are also incorporated into the ONIX lists, but any ISO deletions are only deprecations in ONIX.

Did you know…

that you can send more than one short or long description of your product, each aimed at a different audience ?

Above, I noted that you can send multiple repeats of textual data, where the same 'text' is sent in multiple languages. But you can also send multiple short descriptions, long descriptions, or any other marketing collateral, each intended for a different audience – so perhaps one description for consumers, a different one for bookstore staff, and a third for librarians. The key to this lies with the <ContentAudience> tag:

```
<TextContent>
    <TextType>02</TextType>
    <ContentAudience>03</ContentAudience>
    <Text>The latest fantasy novel from children’s author Claudia Monteverde.</Text>
</TextContent>
<TextContent>
    <TextType>02</TextType>
    <ContentAudience>02</ContentAudience>
    <Text textformat="05"><ul><li>Published to coincide with promotion around
release of the animated movie</li><li>Point of sale material also available</li>
</ul></Text>
</TextContent>
```

You can see the <TextContent> composite is repeated with the same <TextType>, but each has a different audience [6].

[6] While the example uses <TextContent>, you can do something similar with <CitedContent> and with <SupportingResources> too, though the requirement arises much less often.

The most common Content audience code is 00, which means ‘for everyone’. But it’s perfectly possible to suggest a chunk of text is optimised for end customers (retail purchasers and readers) with code 03, or to people in the book trade with code 02. Of course people in the book trade will in reality have access to both texts, but ideally, a retailer should display only the 00 or 03 text to retail customers on its website, while also displaying the 02 text on its internal systems. Other Content audience codes allow you to target librarians or teachers with specific text too. The ‘messages’ might be subtly different, or the more specific text might contain extra information that’s not relevant to other audiences.

It’s probably best to always include an unrestricted ‘for everyone’ text in addition to any text targeted for a more specific audience. And, as with several other Did you know…? features, data senders should discuss with their data recipients whether they can actually make use of this differentiated text, and exactly how the recipient will present the texts in their various business-facing and customer-facing systems.

EDItEUR
10/09/2019
This Application note is a compilation of material from
a series of posts to the ONIX_implement mailing list and
the *ONIX 3.0 Implementation and Best Practice Guide*