ONIX for Books release 3.0.2

Contents

This document summarises the key additions and changes made in ONIX 3.0 rev. 2. Initial proposals were agreed at meetings of the ONIX International Steering Committee and widely discussed as documentation was developed, and text in this document is taken either from earlier discussion papers or from the latest update of the *ONIX 3.0 Implementation and Best Practice Guide*.

All the changes are entirely 'backward compatible' – any ONIX message conforming to ONIX 3.0 or 3.0.1 also conforms to 3.0.2. Implementers of ONIX 3.0 should be able to handle 3.0.2 messages, even if they make no use of the new data elements that might be included. However, some care may need to be taken with the <SalesRights> and <SalesRestriction> composite – see 1. below.

This release of ONIX 3.0 requires the use of Codelists Issue 24 or later, as a number of new codelists are used by new data elements.

1. Repositioning of <SalesRestriction> within <SalesRights>

The <Territory> composite is used in three contexts in ONIX 3.0: in the statement of overall sales rights in P.21, in the statement of the extent of a market within which a supplier operates in P.24, and within <Price> in P.26 to state those countries where a price is valid.

Two of the <Territory> composites may be associated with <SalesRestriction> composites. However, in earlier revisions of ONIX 3.0, they have different structures:

```
<SalesRights>
    <Territory> … </Territory>
</SalesRights>
<ROWSalesRightsType> … </ROWSalesRightsType>
<SalesRestriction> … </SalesRestriction>
```

and:

```
<Market>
    <Territory> … </Territory>
    <SalesRestriction> … </SalesRestriction>
</Market>
```

In the case of <Market> it is clear that any sales restriction applies within the stated territory (and conversely does *not* apply in other parts of the same market or in other markets altogether). In the case of <SalesRights>, this close linkage is not present. A temporary fix for this was put in place in 2010, with the addition of sales rights type codes 07 and 08, which can be used to link a <Territory> to a sales restriction. However, this fix cannot be used in complex cases where there is more than one sales restriction (*eg* partial exclusivity to a small group of retailers).

ONIX 3.0.3 has adopted a modification the structure of <SalesRights> so that it closely matches the structure of <Market>:

```
<SalesRights>
    <Territory> … </Territory>
    <SalesRestriction> … </SalesRestriction> <!-- new location -->
</SalesRights>
<ROWSalesRightsType> … </ROWSalesRightsType>
<SalesRestriction> … </SalesRestriction> <!-- old structure deprecated -->
```

This clearly links the sales restriction to the territory within which the restriction operates and removes the need for codes 07 and 08. The use of <SalesRestriction> in its original location is deprecated.

The cases where this change adds significant flexibility and semantic precision are quite rare. For any data supplier currently making use of the <SalesRestriction> composite, this should be a very simple restructuring of the message. For data recipients, a little more development work would be required. However, with the increasing importance of e-books, there are more products that are available via retailer-specific platforms, and more and more complex channel-based sales restrictions, and increased clarity here will be helpful.

2. New license information for e-publications

ONIX 3.0 includes a <UsageConstraint> composite that can summarise certain types of licensing conditions under which an e-book can be used – for example whether the text can be cut and pasted, or whether it can be ‘lent to a friend’ (such possibilities may be enabled or disabled by the reading platform too). But the ONIX cannot include a full statement of the licensing terms.

For certain types of publishing, this is critical. A new <EpubLicense> composite has been introduced to deliver details of the license terms for a digital product. The license must have a name or title, and if the license is available on the internet, a link to the actual license may be provided. There may be several links to different expressions of the same license – for example a link to a legal document for specialist use, and a separate link to a summary intended for consumers – and links to ‘machine-readable’ license expressions such as ONIX-PL or CCrel are likely to become more valuable in the future (particularly in library contexts).

```
<EpubLicense>
    <EpubLicenseName>Creative Commons Attribution 4.0 International Public License
</EpubLicenseName>
    <EpubLicenseExpression>
        <EpubLicenseExpressionType>01</EpubLicenseExpressionType>
        <EpubLicenseExpressionLink>http://creativecommons.org/licenses/by/
         4.0/deed</EpubLicenseExpressionLink>
    </EpubLicenseExpression>
    <EpubLicenseExpression>
        <EpubLicenseExpressionType>02</EpubLicenseExpressionType>
        <EpubLicenseExpressionLink>http://creativecommons.org/licenses/by/
         4.0/legalcode</EpubLicenseExpressionLink>
    </EpubLicenseExpression>
</EpubLicense>
```

This example shows a Creative Commons license (CC-BY 4.0) is applied to an e-book, and provides two links, one the URL of the easy-to-read license summary, and the other to the full legal document.

3. New <NoPrefix/> flag in titles

All earlier versions of ONIX have allowed the provision of either a <TitleText> element (including its equivalent in ONIX 1.x), or a combination of <TitlePrefix> and <TitleWithoutPrefix>. For languages where definite and indefinite articles such as 'A' or 'The' are usually ignored for title sorting purposes, this allows data suppliers to differentiate clearly the part of the title to be ignored.

Unfortunately, limitations in some legacy data supply systems mean that not every data supplier can make use of <TitlePrefix> – and this inevitably means that, for titles supplied using <TitleText>, there are two possibilities:

a. there is no non-sorting prefix;
b. the supplier of the metadata is unable to separate out any prefixes.

In turn, data recipients who *do wish to differentiate* must inspect every record that uses <TitleText>.

The introduction of a <NoPrefix/> empty element is intended to separate possibilities a) and b), and to enable a data supplier provide a positive indication that there is no prefix that should be ignored for sorting purposes.

The flag has been introduced as an alternative to <TitlePrefix>. So for titles which definitely lack a prefix, the new combination of <NoPrefix/> and <TitleWithoutPrefix> is now valid. For titles which definitely include a prefix, <TitlePrefix> and <TitleWithoutPrefix> should continue to be used, and for systems that cannot distinguish between titles with or without prefixes, <TitleText> should be used for all titles as before. Examples of the three might look like this:

Title known to *include* a non-sorting prefix:

```
<TitleElement>
    <TitleElementLevel>01</TitleElementLevel>
    <TitlePrefix>The</TitlePrefix>
    <TitleWithoutPrefix>man on the balcony</TitleWithoutPrefix>
</TitleElement>
```

Title known *not to include* a non-sorting prefix:

```
<TitleElement>
    <TitleElementLevel>01</TitleElementLevel>
```

```
    <NoPrefix/>
    <TitleWithoutPrefix>Roseanna</TitleWithoutPrefix>
</TitleElement>
```

Title for which the data supplier *cannot differentiate* between titles with and without prefixes:

```
<TitleElement>
    <TitleElementLevel>01</TitleElementLevel>
    <TitleText>The laughing policeman</TitleText>
</TitleElement>
```

This approach maintains the current logic for recipients – that is, any <TitleText> received has to be inspected because the supplier of the data has not stated (or been able to state) whether there is a prefix or not. But as ONIX users begin to implement the <NoPrefix/> flag, the number of records that require such intervention should decrease.

4. Including city or town names within <ContributorPlace>

A means to associate a contributor with a country and/or region was introduced into ONIX 2.1 revision 02. Such connections can provide powerful 'local author' promotion options. In ONIX 3.0, this was generalised so that multiple connections (born in…, currently resides…) can be listed.

The <ContributorPlace> composite in earlier versions of ONIX 3.0 looks like this:

```
<ContributorPlace>
    <ContributorPlaceRelator>04</ContributorPlaceRelator>
    <RegionCode>CA-NL</RegionCode>
</ContributorPlace>
```

This indicates the contributor currently resides in Newfoundland and Labrador. Note that <CountryCode> can be used instead of <RegionCode>. They can also be used together, though this is not recommended because all region codes already contain an indication of the relevant country.

From revision 3.0.2, a new <LocationName> element is added to the composite, so the structure consists of either (or both) of <CountryCode> and <RegionCode>, followed by an optional <LocationName>. Typically, the place name should be a town or city, and should never repeat the country or region name:

```
<ContributorPlace>
    <ContributorPlaceRelator>04</ContributorPlaceRelator>
    <RegionCode>CA-NL</RegionCode>
    <LocationName>Killick-Claw</LocationName>
</ContributorPlace>
```

Geographical names are also subject to language and script variations (*eg* Milano, Milan, Милан), and <LocationName> is repeatable if necessary. Like other data elements that can be repeated with parallel text in multiple languages, if <LocationName> is repeated, a *language* attribute is required.

5. New <Proximity> and <Velocity> elements within <Stock>

When reporting stock quantities held within the supply chain (at a distributor or wholesaler), the <Stock> composite can contain details of copies on hand and available to fulfil new orders, copies on order, and any committed back order quantity (*ie* copies on order but which will not be available to fulfil

new orders). However, many organisations have policies that prevent them revealing exact stock quantities, as these can be commercially sensitive.

Within EDItX messages – specifically the Trade Stock report message – an optional <Proximity> element may qualify the on hand, on order and committed quantities, stating whether the quantity is exact, approximate, or simply a minimum or maximum figure. Similarly, a <Velocity> composite can be included to specify the rate of depletion of stock.

These options have been added to ONIX 3.0.2, which should allow stockholding intermediaries to report approximate physical stock figures more easily as an enhancement to basic product availability information, and ensures ONIX 3.0 can carry the equivalent information that might otherwise require a separate EDItX Stock report, or EDI message (*eg* a Tradacoms, EDIFACT or X12 standard stock report):

```
<Stock>
    <LocationName>Carlisle</LocationName>
    <OnHand>1400</OnHand>
    <Proxmimity>06</Proximity>
    <Velocity>
        <VelocityMetric>01</VelocityMetric>
        <Rate>125</Rate>
        <Proximity>05</Proximity>
    </Velocity>
</Stock>
```

This indicates that there are *at least* 1400 copies on hand, giving potential customers confidence that even a large order could be fulfilled in full, but also indicates that the stock is being depleted by about 125 copies per day on average, so more than a few days delay in placing an order may result in part-fulfillment or a backorder.

6. Including a <ProductIdentifier> within <PriceCondition>

There are an increasing number of publishers and retailers bundling products together, for example bundling the paperback and e-book versions of a book together in various ways.

If this is a true ‘bundle’, it in effect creates a third product, one with two component parts – so there would probably be a paperback, an e-book and a bundle consisting of both available in the market, each of which should have a conventional ONIX product record of its own. The record for the bundle would include two <ProductPart> composites. But publishers and retailers are experimenting with other methods, for example ‘buy the e-book, get the paperback at a reduced price’ (or *vice versa*). This means that there are just two products (not three). At least one of those products has two prices – one price being conditional on the prior or simultaneous purchase of the other product.

To describe this in ONIX 3.0, <PriceCondition> must be able to specify the linked product that qualifies the purchaser for the reduced price. So in revision 3.0.2, <PriceCondition> is extended to include a <ProductIdentifier> composite:

```
<PriceCondition>
    <PriceConditionType> … </PriceConditionType>
    <PriceConditionQuantity> … </PriceConditionQuantity>
    <ProductIdentifier> … </ProductIdentifier>
</PriceCondition>
```

In the case of a paperback / e-book price offer, the paperback price could be $9.95 and the e-book $7.95, but an additional <Price> composite in the product record for the e-book could carry a $3.95 price conditional on prior purchase of the paperback.

```
<Price>
    <PriceType>01</PriceType>
    <PriceCondition>
        <PriceConditionType>00</PriceConditionType>
    </PriceCondition>
    <PriceAmount>7.95</PriceAmount>             <!-- standard price $7.95 -->
    …
</Price>
<Price>
    <PriceType>01</PriceType>
    <PriceCondition>
        <PriceConditionType>05</PriceConditionType> <!-- based on prior purchase -->
        <ProductIdentifier>
            <ProductIDType>03</ProductIDType>
            <IDValue>9780007120765</IDValue>
        </ProductIdentifier>
        <ProductIdentifier>
            <ProductIDType>15</ProductIDType>
            <IDValue>9780007120765</IDValue>
        </ProductIdentifier>
    </PriceCondition>
    <PriceAmount>2.95</PriceAmount>              <!-- offer price $2.95 -->
    …
</Price>
```

7. New <PriceIdentifier> for sales reporting purposes

Increasingly complex pricing is a feature of the e-book world, with different prices set for the same product via different retailers, different prices for different customer groups even via the same retailer, and different prices for different rental durations for the same ‘product’ (or at least, rentals of different duration under the same product identifier). Reporting these sales via – for example – the EDItX Sales report format becomes less easy to interpret. A single retailer may be reporting on sales of a single product at several different prices.

A simple <PriceIdentifier> composite has been added as an element within <Price>. This will always be a ‘proprietary’ identifier, assigned by the publisher, distributor or wholesaler. The identifier simply labels a particular combination of <PriceAmount>, the terms and conditions expressed in <PriceType>, <PriceQualifier> and <PriceCondition>, currency, country, tax details *etc*. The new label can then be specified at line level within a sales report, to allow highly granular reporting on sales of the product.

> NB This is not the same as the <PriceCoded> composite. <PriceCoded> is an alternative to <PriceAmount>, and identifies a particular price point (eg £6.99, £8.99…) rather than a full set of price, terms and conditions, currency *etc*.

The composite has the usual ONIX structure for an identifier, with the caveat that since the identifier type is always ‘proprietary’ (code 01), the <IDTypeName> element is effectively mandatory.

The <PriceIdentifier> composite can carry a label that is:

- unique for each price amount (*ie* two unrelated products both priced at €4.95 would carry the same identifier);
- unique for each price type (*ie* two unrelated products sold with the same combination of price type and qualifier, conditions, currency, territory *etc* – but not necessarily at the same actual amount – would carry the same identifier);
- unique for each combination of price amount and type (but two unrelated products may still carry the same price identifier if all details of their prices are identical);
- or unique across all products, price points and types.

Whether the ID identifies the price amount, the price type or some combination of both will depend on the choice made by the publisher or supplier. But the price identifier sent in the ONIX message can always be included in a subsequent revenue report to ensure that the publisher or intermediary supplier receiving the revenue report can associate revenue with a particular product sold under particular terms and conditions.

The new composite can be used like this, to add labels to two prices for the same product:

```
<Price>
    <PriceIdentifier>
        <PriceIDType>01</PriceIDType>
        <IDTypeName>PRH PTUID</IDTypeName>
        <IDValue>ff8a5521-fcad-4a3b-89b0-34f2ed131016</IDValue>
    </PriceIdentifier>
    <PriceType>02</PriceType>
    <PriceQualifier>05</PriceQualifier>
    <!-- … details of consumer price inc-tax -->
</Price>
<Price>
    <PriceIdentifier>
        <PriceIDType>01</PriceIDType>
        <IDTypeName>PRH PTUID</IDTypeName>
        <IDValue>3e304b73-50f6-432d-8e82-2865198dad73</IDValue>
    </PriceIdentifier>
    <PriceType>01</PriceType>
    <PriceQualifier>06</PriceQualifier>
    <!-- … details of institutional price exc-tax -->
</Price>
```

In this case, the supplier has chosen to use a UUID as the price identifier, and UUIDs make good labels as they can be minted on demand and completely automatically.

> Note that an equivalent addition will need to be made within the EDItX Sales report format to carry reporting by price identifier.

8. New <NoProduct/> flag

Some ONIX data suppliers provide update (‘delta’) files on a regular schedule. If that schedule is very frequent, perhaps daily, there are on occasion times when there is no update to send – because there has not been any change to the data since the last update. This can cause some confusion for the data recipient, who is expecting the usual daily update.

To avoid this, a new <NoProduct/> flag has been added. The sole valid use of this is to denote an 'empty message' within a stream of updates sent on a prearranged timetable, providing a positive indication that there have been no changes since the previous update. The message must contain *either* <NoProduct/> *or* one or more <Product> composites.

For example, a minimal 'no changes' message might look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ONIXMessage release="3.0">
<Header>
    <Sender>
        <SenderName>GlobalBookinfo</SenderName>
    </Sender>
    <MessageNumber>231</MessageNumber>
    <SentDateTime>20140125T1115-0500</SentDateTime>
</Header>
<NoProduct/>
</ONIXMessage>
```

A message like this ensures that a pre-arranged timetable or predictable sequence of message numbers can be maintained, even when there are no changes that need to me communicated.

9. More flexible discounts

A new <ToQuantity> element has been added within <Discount>, and initial values added List 170 used by <DiscountType>, to allow more flexible descriptions of discount structures.

When considering a 'rising discount', a structure like this suggests that 20% discount is given irrespective of order size, rising to 30% on orders for 11 or more copies:

```
<Discount>
    <DiscountPercent>20</DiscountPercent>
</Discount>
<Discount>
    <Quantity>11</Quantity>
    <DiscountPercent>30</DiscountPercent>
</Discount>
```

In general, it is assumed that the 30% applies to all copies in the order, and this is the default if <ToQuantity> is omitted. However, it is also possible that the higher discount applies only to the 11th and subsequent copies – so if 15 copies are ordered, the first ten attract a discount of 20% and the last 5 attract a discount of 30%. In ONIX 3.0.2, this can be described like this, with an upper limit on the number of copies that attract a particular percentage discount:

```
<Discount>
    <DiscountType>03</DiscountType>
    <Quantity>1</Quantity>
    <ToQuantity>10</ToQuantity>
    <DiscountPercent>20</DiscountPercent>
</Discount>
<Discount>
    <DiscountType>03</DiscountType>
    <Quantity>11</Quantity>
    <ToQuantity>0</ToQuantity>
    <DiscountPercent>30</DiscountPercent>
</Discount>
```

Note that the presence of <ToQuantity> should not be relied upon to signal such a 'progressive' discount – List 170 has been populated and code 03 should be used with <DiscountType>.

It is also possible to arrange discounts so that the percentage depends on the number of copies ordered *previously*, over some specific time period, rather than simply on the number of copies in a *particular* order. And occasionally, subsequent orders can in a specific time period can also affect the discount. This can result in *retrospective* discounts being applied or claimed. New code values in List 170 can be used to make these distinction clear.

10. Other minor changes

- A new <CopyrightType> element has been added to the <CopyrightStatement> composite, to allow statements about phonogram rights for audiobooks.
- A new <PrizeStatement> element has been added within <Prize> to accommodate text or terminology that is specific to particular prizes or awards – for example some Prize recipients are known officially as 'winners', others as 'medalists', 'laureates' and so on. The <PrizeStatement> should be used for display purposes only, alongside coded information in the established data elements. Thus the <PrizeCode> element should be used to select all 'winners', even if officially they might be displayed as 'medalists'.
- The <Complexity> composite is no longer deprecated, and should be used for quantitative and objective measures of text complexity such as Lexile.
- An explicit <SupplierCodeTypeName> has been added to the <SupplierOwnCoding> composite.
- There are a number of changes of cardinality to support repeated elements carrying parallel text in multiple languages., and new *language* and *textscript* attributes added to some affected elements.

11. Naming and migration

These changes constitute Revision 2 of ONIX for Books 3.0 (also known as 3.0.2). All current ONIX messages remain valid under the revised schema, so there is no requirement to maintain original and updated schemas in parallel – all ONIX 3.0 users should update to the latest documentation, and data providers should update to the latest schema files and codelists for validation purposes even if they do not intend to use any of the new features introduced in 3.0.2.

Data recipients should update to the latest schema files as soon as practicable, even if they cannot immediately process any of the new data elements or attributes – they should for the most part simply ignore any elements they cannot make use. An important exception is that a <SalesRestriction> placed in the new location within <SalesRights> should not be ignored.

Graham Bell
EDItEUR
24th January 2014