

ONIX for Books

Product Information Message

Application Note: Sales Rights in ONIX 3.0

The section of ONIX that defines the sales rights for a particular book is a vital piece of the metadata puzzle, but is one that causes frequent headaches. Part of the problem is undoubtedly that publishers often have relatively poor data about rights (or more specifically, while they have good knowledge of their publishing rights, they often lack sales rights data in a structured form – or they have comprehensive sales rights data, but held in a system that doesn't integrate with the product management system that produces their ONIX.

And it isn't helpful that there has been conflicting best practice advice in the past, and that ONIX 2.1 is notably inconsistent when it comes to describing geographical areas for sales rights, markets, suppliers and so on.

Defining geographical areas

In ONIX 3.0, the specification of sales rights centres around the <Territory> composite – the composite defines a geographical area, across which the sales rights are uniform and set to one of the available sales rights types.

Within <Territory>, you can **define a geographical area through addition** – by building up a list of countries and regions that are included in the area. So:

```
<Territory>
  <CountriesIncluded>CA MX US</CountriesIncluded>
</Territory>
```

defines a geographical area covering the three North American countries *added together*¹.

You can also combine *parts* of countries:

```
<Territory>
  <CountriesIncluded>US MX</CountriesIncluded>
  <RegionsIncluded>CA-NB CA-NL CA-NS CA-PE</RegionsIncluded>
</Territory>
```

Here the area is the combination of the whole of Mexico and USA, plus just four Canadian provinces. Somewhat obviously, the regions included must be parts of countries that have not *already* been included. Including US and then also including US-AK (Alaska) makes no sense.

Within <Territory>, you can also **define a geographical area through subtraction**. Here's an example:

```
<Territory>
  <CountriesIncluded>CA MX US</CountriesIncluded>
  <RegionsExcluded>CA-QC</RegionsExcluded>
</Territory>
```

where the territory consists of the whole of North America (three countries added together) but then *minus* Quebec. In this case, the regions excluded *must be parts of countries that have already been included*. Including Mexico, USA and Canada and then excluding region CN-53 (Yunnan, a province of China) makes no sense.

The Quebec example subtracts a region. You can also use subtraction of *whole countries* to define a territory – though this is a special case, as you can only subtract whole countries from a region called 'WORLD'. WORLD is the only region that is bigger than a country⁴, and it can only be used in <RegionsIncluded>:

```
<Territory>
  <RegionsIncluded>WORLD</RegionsIncluded>
  <CountriesExcluded>CA MX US</CountriesExcluded>
</Territory>
```

Here you start by adding the whole world, then subtracting the three countries, so the territory consists of 'everywhere outside North America'.

Within <Territory>, there are four possible country and region tags – two for addition and two for subtraction:

<CountriesIncluded>
 <RegionsIncluded>
 <CountriesExcluded>
 <RegionsExcluded>

But as ever with ONIX, the order of the various XML tags matters:

- inclusions always come before exclusions
- countries come before regions

and only certain combinations of the four tags make sense:

- <CountriesIncluded> for adding countries
- <RegionsIncluded> for adding regions
- RI with CX but only where RI = WORLD – a whole country cannot be subtracted from a region that is just part of a country

For many publishers, and for publishing in most languages, these three simple cases are enough – they are by far the most common. There are five other combinations that make sense:

- RI with RX but only where RI = WORLD
- RI with CX *and* RX only where RI = WORLD, and the excluded regions are not parts of the countries already excluded) ²
- CI with RX the excluded regions must be parts of the included countries
- CI with RI only where RI does not contain WORLD, and the included regions are not already part of the included countries
- CI with RI *and* RX only where RI does not contain WORLD, and the excluded regions are parts of the countries already included) ³

Given all this, you should be able to see that a territory that's comprised of 'North, South and Central America' will consist of a list of countries in <CountriesIncluded>. The *complementary territory* 'everywhere except the Americas' will consist of a list of countries – in fact the *same* list of countries – in <CountriesExcluded> that are subtracted from <RegionsIncluded> WORLD. Something complex like 'Francophone markets' might consist of a list of mostly European and African <CountriesIncluded>, plus an additional list of <RegionsIncluded> (for example Quebec). The most complex sort of combination is the complementary option 'everywhere except Francophone markets', which would be most effectively listed as <RegionsIncluded> WORLD, with a list of countries – the *same* list of mostly European and African countries – but switched into <CountriesExcluded> and <RegionsExcluded> for Quebec.

This requirement for '**complementary territories**' is fairly common – in one territory you have one type of sales rights, and everywhere else, you have some other type of sales rights. And typically, one territory would be a list of additions, and the second would be a list – the *same* list – subtracted from WORLD.

Setting the rights in a territory

So first, **what are sales rights?** When publishers discuss rights, they are in fact usually talking about publishing rights – the rights they acquire directly or indirectly from the author – or perhaps 'subrights', subsets of the publishing rights that they sell on to other publishers. Sales rights are different, and they are the rights the publisher passes along to their distributors, wholesalers and retailers.

Fundamentally, publishing rights concern *where a publisher has the right to sell a product*. The sales rights are *where the publisher chooses to make the product available* (ie chooses to **exercise** those publishing rights). Publishing rights generally apply to a *work*, whereas sales rights apply to a *product*. Clearly the sales rights must be a subset of (or the same as) the publishing rights. And in ONIX for Books, only the sales rights are described in detail.

Looking at sales rights, at their simplest, **a product is either for sale in a particular territory, or it is not for sale** – but there are a few nuances on top of that. Sales rights type codes 01 and 02 mean the product is for sale in the specified territory. Sales rights type codes 03, 04, 05 and 06 mean the product is not for sale in the specified territory. So:

```

<SalesRights>
  <SalesRightsType>03</SalesRightsType>
  <Territory>
    <CountriesIncluded>CA MX US</CountriesIncluded>
  </Territory>
</SalesRights>

```

simply means ‘the product is not for sale in North America (CA, MX, US)’.

It’s important to understand that the ‘not for sale in North America’ statement implies nothing about whether or not the product is for sale in Europe or anywhere else. It means *only* ‘not for sale in North America’. And given that bookselling is a highly international business, it is very much a **good practice to provide complete sales rights information**: you should try to provide the sales rights everywhere else as well. Here’s where the *complementary territories* discussed above come in:

```

<SalesRights>
  <SalesRightsType>01</SalesRightsType>
  <Territory>
    <RegionsIncluded>WORLD</RegionsIncluded>
    <CountriesExcluded>CA MX US</CountriesExcluded>
  </Territory>
</SalesRights>
<SalesRights>
  <SalesRightsType>03</SalesRightsType>
  <Territory>
    <CountriesIncluded>CA MX US</CountriesIncluded>
  </Territory>
</SalesRights>

```

This example uses two complementary territories, and it says ‘for sale everywhere except North America’ and ‘not for sale in North America’. Between the two <SalesRights> composites, the whole world is covered.

It doesn’t matter what order the two <SalesRights> composites come in, but it’s conventional to put the positive ‘for sale’ composites first.

The nuances are all about **the relationship between these sales rights and the *publishing rights*** – the underlying work rights that the sales rights are derived from. There are six options for

<SalesRightsType>⁵:

- 01 Product is for sale, based on the publisher holding exclusive publishing rights
- 02 Product is for sale, based on the publisher holding only non-exclusive publishing rights⁶
- 03 Product is not for sale (for any reason)
- 04 Product is not for sale (the publisher holds exclusive publishing rights, but has chosen not to exercise them⁷)
- 05 Product is not for sale (the publisher holds non-exclusive publishing rights, but has chosen not to exercise them)
- 06 Product is not for sale (because the publisher does not hold the relevant publishing rights at all)

The nuances can help retailers. Code 04 is more meaningful than code 03, and it avoids the retailer having to search for similar products from another publisher – there can’t be any, because the publisher in question has exclusive rights to publishing the work. Similarly, code 06 means ‘although this product is not for sale in this particular territory, it’s because someone else holds the rights – and so it *is* worth searching for another very similar product’. Equally, codes 04, 05 and 06 can be used to help detect infringements of publishers’ territorial rights, where code 03 cannot.

Ensuring global coverage

With relatively simple cases, it's easy for a publisher to work out whether the sales rights information provided is complete. Are the rights so simple they use just one WORLD territory with no exclusions? If so, nothing more is needed. But in more complex cases, or cases where the publisher's own data is incomplete, ONIX 3.0 *requires* that the sales rights are completed by *adding an* `<ROWSalesRightsType>` *element*.

So you can say, 'for anywhere I haven't specified, the product is not for sale':

```
<ROWSalesRightsType>03</ROWSalesRightsType>
```

The sales rights type codes 01–06 are all possible options with these rest of world rights, but it is more common and somewhat easier to understand if you limit it to the not-for-sale codes 03–06. It is not good practice to use `<ROWSalesRightsType>` with 01 or 02.

How do you work out whether `<ROWSalesRightsType>` is required? There is a simple three-step test:

1. is the WORLD region included in a sales rights composite?
 - if no, `<ROWSalesRightsType>` is always required, even if you believe you have exhaustively listed all the ISO country codes separately. It will act as a fallback if you have inadvertently omitted a country, or if a new country code is added to the ISO list;
 - if yes, see question 2;
2. within the `<SalesRights>` composite that includes WORLD, are there any `<CountriesExcluded>` or `<RegionsExcluded>` elements?
 - if no, `<ROWSalesRightsType>` must be omitted;
 - if yes, see Question 3;
3. are all those excluded countries and regions *included* in other `<SalesRights>` composites?
 - if no, `<ROWSalesRightsType>` is required. It may be used to specify the rights that apply in the remaining countries or regions that you left out;
 - if yes, `<ROWSalesRightsType>` must be omitted.

And there is one final special case with `<ROWSalesRightsType>`. Using code 00, it is possible to say 'for anywhere I haven't specified, I have no idea what the sales rights are'. **Code 00 means 'sales rights are unknown or unstated'**. This is useful, because the publisher's own data is often incomplete. Cautious data recipients should of course treat code 00 as a 'not for sale' code. Note that Code 00 cannot be used in an ordinary `<SalesRights>` composite.

Of course, this means that there are two ways of stating the same sales rights:

```
<SalesRights>
  <SalesRightsType>01</SalesRightsType>
  <Territory>
    <RegionsIncluded>WORLD</RegionsIncluded>
    <CountriesExcluded>CA MX US</CountriesExcluded>
  </Territory>
</SalesRights>
<SalesRights>
  <SalesRightsType>03</SalesRightsType>
  <Territory>
    <CountriesIncluded>CA MX US</CountriesIncluded>
  </Territory>
</SalesRights>
```

and:

```
<SalesRights>
  <SalesRightsType>01</SalesRightsType>
  <Territory>
    <RegionsIncluded>WORLD</RegionsIncluded>
    <CountriesExcluded>CA MX US</CountriesExcluded>
  </Territory>
</SalesRights>
<ROWSalesRightsType>03</SalesRightsType>
```

are both complete and correct, and mean exactly the same thing. Is one preferred over the other? Well, it mostly comes down to a personal view. The only difference is that the first is slightly more 'definitive' in tone, and leaves room for adding an <ROWSalesRightsType> 00 later if necessary.

Common issues

The most common and most serious issue is that sales rights information is missing altogether from a publisher's ONIX records. Yet this is vital information for distributors and retailers. In the absence of any information, a retailer simply cannot tell whether they have the right to offer the product for sale in any particular country. Assumptions based on the country in which the publisher is based, or upon the currency of any price, can be misleading.

Some minor issues arise because of the fact that certain ISO 'country codes' in fact identify small territories or *parts* of countries that have some special administrative status, partial independence or are geographically distant from the larger country. The code CN for China does not cover Hong Kong or Macau, which carry their own codes. The code for France covers only metropolitan France, as French overseas departments and territories such as Réunion, French Guiana or Guadeloupe have their own codes. These codes should always be included separately in the relevant country code list, in addition to the 'main' country code.⁸

There are a handful of country parts that have both their own country code and a region code within that country. The obvious example would be the UK Isle of Man (country code IM, region code GB-IOM). Always use the country code – these region codes are deprecated.

Special attention should be given to the requirement for <ROWSalesRightsType>. This is intended to eliminate most issues that arise from retailers being forced to make assumptions – it allows publishers to specify the 'default' rights that apply in any country not specifically mentioned. However, there is still an unavoidable issue when code 00 is used in <ROWSalesRightsType>, meaning the sales rights are unknown or unstated. In this instance, a cautious retailer should assume the product is not for sale in the affected countries.

Finally, sales rights can change throughout the active commercial life of a product, for example through a change in the publisher's distribution strategy, or through changes in the underlying contractual rights held by the publisher. The <SalesRights> in ONIX are subject to updates at any time.

How does ONIX 2.1 compare?

Defining sales rights in 2.1 and 3.0 are quite different – there is no 2.1 sales rights equivalent of <CountriesExcluded> or <RegionsExcluded> and 'territories' can only be constructed additively. And there is no direct equivalent of <ROWSalesRightsType> – but this can be simulated using a final <SalesRights> composite with the relevant rights type and <RightsTerritory> = ROW⁹.

Territories in ONIX 3.0 are consistent – they are constructed the same way, using the <Territory> composite, in <SalesRights>, in <TextContent> and elsewhere in Block 2, in <Market> and in <Price>. In contrast, ONIX 2.1 uses four entirely different structures to do equivalent jobs.

Notes

1. ignoring (for now) any oddities caused by ISO country codes that cover small parts of countries, like PM (St. Pierre and Miquelon, a pair of small islands just off the coast of Newfoundland, which are 'in' North America but are in fact part of France)
2. this is a pretty rare case, but it means that both the countries and the regions are (separately) excluded from WORLD. It does not mean that the countries have the regions excluded, and then the remaining parts of the countries are excluded from WORLD (which would leave the regions as part of WORLD)
3. this would be hugely more complex if some regions were parts of other regions. With the exception of WORLD, regions in List 49 are not nested in this way – so in effect the excluded regions must be parts of the included countries. See also note 8 below.
4. ignoring ECZ, the Eurozone. The code is kept in ONIX list 49 solely for backward compatibility, and should never be used.
5. options 07 and 08 were added as a temporary workaround that is now unnecessary. They should not be used.
6. non-exclusive publishing rights mean that another publisher also has the right to exploit the work in the same territory. It has nothing to do with the product being exclusive to a particular retailer, or any exclusive distribution arrangement.
7. or at least, has chosen not to exercise them with this particular product. It doesn't rule out exercising those rights with some other product. The most common example of this is where a publisher holds global exclusive rights. They publish a hardback, made available in territory A and a paperback everywhere else (territory B). The sales rights for the hardback would be type 01 in territory A and 04 in territory B. The sales rights for the paperback would be the opposite. Work identifiers can be immensely useful in such situations, though if a good work identifier is unavailable, <RelatedProduct> should be used to link the two.
8. This interpretation – that the country code for a country does not cover any parts of that country that have their own country codes – is a key part of how ONIX works. List 91 should always be used this way. Other standards may interpret the underlying ISO 3166-1 country code list differently, but the ISO list acknowledges that many (more than 50) of its country codes are not independent or sovereign states. The ONIX list is *based on* the ISO list, but the special ONIX interpretation ensures that any location is covered by only a single and unambiguous country code – and like a jigsaw, the geographical areas represented by each code can be fitted together without any overlaps.
9. ROW is a code in list 49, but it may not be used in ONIX 3.0 (and has in fact been removed in recent versions of the codelists that are intended exclusively for use with 3.0).

EDItEUR
22/12/2016