ONIX for Books 3.0: future modification to XSD and RNG schema files

This document provides a warning of some improvements that will be made in the future to the ONIX 3.0 XSD and RNG schema files. These improvements to the schema files will not require any changes to properly-constructed ONIX files.

ONIX 3.0 XSD and RNG schemas

Preparations have been made for some improvements to the schema files used for message validations and by applications that use the schema structure in constructor functions to create new record or message object instances. The upcoming schema file changes will all be 'transparent' for properly-constructed messages and do not imply any change to the ONIX specification itself – they merely ensure that validation checks ONIX message files a little more rigorously.

No updates have been made to the official XSDs or RNGs. However, the changes will be implemented in a few months time (no earlier than October 2015), and developers may wish to begin using the preview 'strict' versions of the schema instead of the standard XSD and RNG files. There is no preview of a strict DTD, as DTD validation cannot check data content.

Within the strict XSD and RNG files, numeric data elements are mostly limited to positive numbers, so you cannot have a negative edition number or a zero pack quantity, for example.

Integers

Positive integer (1, 2, 3...)
```
<BatchQuantity>
<ConferenceNumber>
<EditionNumber>
<FreeQuantity>
<LatestReprintNumber>
<MapScale>
<MessageNumber>
<MessageRepeat>
<MinimumOrderQuantity>
<NumberOfCopies>
<NumberOfItemsOfThisForm>
<NumberOfPages>
<PackQuantity>
<PositionOnList>
<SequenceNumber>
```

Positive integer or zero (0, 1, 2, 3...)
```
<CBO>
<Number>
<NumberOfIllustrations>
<OnOrder>
<OrderTime>
```

Integer (…-3, -2, -1, 0, 1, 2, 3…)
```
<OnHand>
<Rate>
```

Real numbers

Positive decimal (> 0)
```
<ExtentValue>
<Measurement>
<PriceAmount>
<TaxableAmount>
```

Positive decimal or zero (≥ 0)
```
<DiscountAmount>
<Quantity>
<TaxAmount>
<ToQuantity>
```

Decimal
```
-none-
```

Percent decimal (≥ 0 and ≤ 100)
```
<DiscountPercent>
<Percent>
<TaxTatePercent>
```

The most common error that stricter validation will catch is the use of a zero price for a free-of-charge item (you should use <UnpricedItemType> instead). More generally, it will catch the use of zero on some elements when the real meaning is 'unknown' or 'not applicable' (where instead, the element should be omitted entirely).