



Jointly with Book Industry Study Group, New York, and
Book Industry Communication, London



ONIX for Books

Product Information Message

Proposals for Release 2.1 Rev.04 (JP)

March 2011

© 2010, 2011 EDItEUR. All rights reserved.

EDItEUR (www.editeur.org) is the international group which coordinates the development and promotion of standards for electronic commerce in the book and serials sectors. EDItEUR has developed the ONIX for Books standard jointly with Book Industry Study Group (BISG), New York, and Book Industry Communication (BIC), London.

Contact EDItEUR by [email](#).

Table of Contents

Introduction	4
1. Proposed Optional Extensions to the ONIX 2.1 Schema	5
1.1 Identification of the language and script used in an element or message	5
1.2 Incorporation of non-ASCII characters in textual data elements	5
1.3 Incorporation of ruby annotations in textual data elements	6
1.4 Supply of collation information for a textual data element	7
1.5 Provision of contact information for individual products	8
1.6 Publication order of a series	9
1.7 Cut-off date for reservation orders	9
1.8 Sets where items are 'Not for individual resale'	9
2 Additions to Codelists (included in Issue 12, October 2010, and in Issue 13, March 2011)	9
2.1 <MainSubject> and <Subject>	9
2.2 <NameCodeType>	10
2.3 <ProductFormDetail>	10
2.4 <PriceQualifier>	10
3 Arrangements for validation of ONIX 2.1 rev.04 messages	11
Appendices	12
Appendix 1 Cut-off date for reservation orders	12
Appendix 2 Sets where items are 'Not for individual resale'	13

Introduction

This document summarises a range of proposed additions to ONIX for Books Release 2.1, which if accepted, and after any necessary revision, will be incorporated into Release 2.1 Revision 04 (JP)[†]. These proposals reflect requests made to EDItEUR by Japanese book publishers and the relevant Japanese trade association. Comments on these proposals are welcome from any national group.

These changes are all additions to the existing, widely deployed ONIX 2.1, and are *specific to those involved in the Japanese book supply chain*. Existing ONIX 2.1 users are advised to continue using ONIX 2.2 Rev 02 or Rev 03 as appropriate. (The latter revision incorporates a number of optional elements specific to supply chain requirements in Spain and Australia, in the same way that this version is specific to Japanese requirements.)

Similar changes are likely to be applied to ONIX 3 at some future date to ensure that ONIX 3 is suitable for wide international application, but are not being proposed at this point – experience gained with these extensions in ONIX 2.1 in Japan will inform future ONIX 3 extensions.

A number of additions that have already been incorporated into ONIX for Books Codelists Issues 12 and 13 are also particularly relevant to the Japanese book trade.

Graham Bell
Chief Data Architect
EDItEUR

graham@editeur.org

[†] This document applies only to ONIX 2.1, except insofar as shared code list values may also affect ONIX 3. Reference names are used for ONIX elements.

1. Proposed Optional Extensions to the ONIX 2.1 DTD and Schema

This proposal concerns a number of optional extensions to the ONIX 2.1 DTD and schema, and the relevant documentation. They are intended only to support the use of ONIX 2.1 within the Japanese book supply chain.

These changes are entirely compatible with the existing schema and would not affect any current ONIX 2.1 records.

1.1 Identification of the language and script used in an element or message

Add a new *textscript* XML attribute that may be attached to any (textual data) element (that is, anywhere the existing *language* attribute may be used). Values for the *textscript* attribute would be the four letter codes taken from ISO 15924 – for example “Latn” for Latin script (as used in the majority of current ONIX), “Jpan” for Japanese script (which may contain Kanji, Hiragana, Katakana, and occasionally, some Latin [Romaji] characters), or “Cyr1” for Cyrillic script. See ONIX codelist 121 (or <http://unicode.org/iso15924/codelists.html>) for the full list of script codes. For example:

```
<TitleText language="jpn" textscript="Jpan">
```

The *textscript* attribute – like the existing *language* attribute – should be used whenever the language or script used in a textual data element differs from the expected language or script of the message.

Note the *language* and *textscript* attributes need only be used where the language or script used varies from the ‘expected’ language or script. It is not necessary in data feeds where the language and script are well known (*ie* within the Japanese book trade, books titled in Japanese, in a mix of Kanji and kana need not carry any attributes, whereas a book titled in English, in Latin script, should carry at least the *language* attribute).

Where the *language* and *textscript* attributes are used on the <ONIXMessage> element, they become the expected language and script for the whole message, and apply to all textual (*ie* non-coded) data and attribute values. In the interests of compatibility, if these attributes are omitted from <ONIXMessage> then English language in the Latin script is the default, except by mutual agreement or understanding between ONIX sender and recipient

Thus, if an expected message-wide language and script are specified using:

```
<ONIXMessage language="jpn" textscript="Jpan">
```

then the *language* and *textscript* attributes do not need to be repeated on each element in the message. The attributes need only be used on individual elements when the language or script in a text element differs from the message-wide expectation.

Note the *xml:lang* attribute defined by the W3C XML 1.0 Recommendation should *not* be used.

1.2 Incorporation of non-ASCII characters in textual data elements

Where the ONIX 2.1 documentation or schema defines text element data content to be limited to ASCII characters, this should be amended to allow the use of any XML-legal characters native to the overall message script, language and character set.

This change specifically affects the following data elements in the header and broadens the range of characters they can include:

<IDTypeName>	<FromCompany>
<FromPerson>	<FromEmail>
<ToCompany>	<ToPerson>
<MessageNote>	

Non-ASCII characters – in the header or throughout the message – should normally be encoded in the encoding scheme used for the message (and defined in an XML encoding declaration), but might also use HTML entities defined in the DTD and schema or XML numerical character references. Thus the message as a whole may be encoded in UTF-8, Shift_JIS, Windows-932, UTF-16 or another encoding, but may additionally include HTML entities and XML numerical character references, and may use a combination of these three methods to incorporate any characters native to the language and script in use.

The character encoding for the message is specified in the initial XML declaration. For a message encoded in UTF-8, it would be:

```
<?xml version="1.0" encoding="UTF-8"?>
```

and UTF-8 is the encoding that is expected to be used within the Japanese book trade.

Note that all ONIX markup tags, and codes such as those defined by the ONIX codelists, should continue to use ASCII characters, as should all data elements that are defined as numeric or as dates. Identifiers and coded data where the identifiers or codes are defined outside of ONIX (eg subject coding schemes) should follow their relevant external authority.

1.3 Incorporation of ruby annotations in textual data elements

Rubies are small glosses providing pronunciation and disambiguation information. In Japanese, they are written in Hiragana or Katakana syllabaries (or occasionally in the Latin alphabet), and are attached to names and unfamiliar terms that are themselves written in Kanji logographic characters. In principle, ruby glosses may be required within any non-coded textual data element.

1.3.1 Incorporation of ruby into textual data elements where XHTML markup is enabled

In ONIX 2.1, XHTML markup is enabled on the following elements:

<Annotation>	<BiographicalNote>
<DownloadCaption>	<DownloadCopyrightNotice>
<DownloadCredit>	<DownloadTerms>
<MainDescription>	<PrizeJury>
<ProductWebsiteDescription>	<ReviewQuote>
<Text>	<TextWithDownload>
<WebsiteDescription>	

To incorporate a ruby gloss in these elements, the established XHTML 1.1 or HTML markup for ‘simple ruby’ should be used. The example given uses XHTML [†]:

```
<Text language="jpn" textscript="Jpan" textformat="05"><p>
  ... <ruby><rb>村上春樹</rb><rp></rp><rt>むらかみ
  はるき</rt><rp></rp></ruby> ... </p></Text>
```

The whole of the annotated phrase is enclosed in a <ruby> element (in blue above). The base Kanji characters are enclosed in an <rb> element (in purple), and the gloss is enclosed within an <rt> element (in red). Optionally, an <rp> element (in green above) can appear both

[†] HTML markup is the same – but as with any other HTML markup, it needs to be enclosed within CDATA tags

before and after the <rt> element – its content appears only if the rendering system cannot properly render the <rt> element as a gloss. This should result in a display like this:

```
    むらかみ はるき
    … 村上春樹 …
```

For further details of ‘simple ruby’, see Sections 1.2.1 and 1.2.2 in <http://www.w3.org/TR/ruby/>. Note that the use of such markup in HTML 4 is not standardised, but does work in many major web browsers. It is standard in XHTML 1.1, and is a proposed standard in HTML 5/XHTML 5. ‘Complex ruby’ is not supported.

1.3.2 Incorporation of ruby into textual data elements where XHTML markup is not enabled

Unicode’s ‘interlinear annotation delimiters’ should be used (see Section 3.6 in <http://www.unicode.org/reports/tr20/>). These can be inserted as numeric character references ￹ (at the beginning of the annotated phrase, in green below), ￺ (to divide the main Kanji from the ruby gloss, in green below), and &#fffb; (at the end of the annotated phrase, in red below):

```
<TitleOfSeries> … &#xfff9;英会話&#xfffa;エイカイワ&#xfffb;シリーズ。 …
</TitleOfSeries>
```

This should result in a display like this:

```
    エイカイワ
    … 英会話シリーズ。 …
```

Note that the gloss only applies to (and in this case is printed above) the first three Kanji characters. These Unicode delimiter characters are for internal use within applications and cannot be rendered by web browsers – they should be converted into the HTML/XHTML equivalent markup before display, if the data is used on a website.

These two methods of embedding ruby glosses are mutually exclusive. Which one should be used depends on whether XHTML markup is enabled for the data element (not on whether the data contains markup other than the ruby gloss).

In order to enable ruby glosses in ONIX textual data elements, a change needs to be made to the DTD and schema files, to allow use of a subset of XHTML 1.1 markup (currently ONIX 2.1 and ONIX 3 incorporate an extract from the XHTML 1.0 schema). Note that the ruby markup is the *only* part of XHTML 1.1 that has been added to the ONIX.

1.4 Supply of collation information for a textual data element

Japanese words such as names are usually written in Kanji, sometimes with a gloss in Hiragana as above. However, collation (sorting) is usually based on a phonetic interpretation of the names. In order to allow proper sorting (*eg* by contributor name or by book or series title), collation information should be supplied using a new *collationkey* attribute:

```
<TitleOfSeries collationkey="エイカイワシリーズ"> …
```

In Japanese text, the collationkey would (usually) be provided using Hiragana.

The following example shows the author name for a book imported into Japan, where the author’s name is ‘John Smith’. The ONIX data contains both a ruby annotation showing how to pronounce the name and a separate collationkey (which is the phonetic equivalent of ‘Smith John’, so that the name can be sorted by family name):

```
<PersonName collationkey="スミス ジョン">&#xfff9;John
Smith&#xffffa;ジョン スミス&#xffffb;</PersonName>
```

Collation keys may in principle be used on any textual data element, but are only of use on elements likely to be used for sorting multiple product records, including series title and title elements and contributor names.

1.5 Provision of contact information for individual products

The requirement here is to allow the name of a contact person to be attached to an individual product, in addition to any contact names that can be included in the Header or in the P.26 Sales Promotion data. The contact in the Header is an appropriate point of contact for issues that concern the message itself or its general content (say, a technical issue), and the contact name in P.26 is specifically for promotional purposes. In contrast, the contact for a product would be a person in an editorial, production, marketing, sales or publicity function for that particular product. All contact names are intended only for use within the trade and must not be disclosed in consumer-facing applications.

Three new data elements inside one new composite element are proposed:

Reference tag	Short tag	Cardinality	Notes
<Contact>	<contact>	0..1	whole composite is optional and non-repeating
<ContactCompany>	<b396>	1	name of the organization represented by the contact, which should always be stated in a standard form. This may be the publisher, or a sales agent, PR agent etc. Mandatory, non-repeating, variable length text, suggested maximum 30 characters
<ContactName>	<b397>	1	free text giving the name, department, phone number etc for a person who is able to answer enquiries from the trade about the product. Suggested maximum length 300 characters
<ContactEmail>	<b399>	0..1	e-mail address for the contact. Optional, non-repeating, suggested maximum length 100 characters
</Contact>	</contact>		

The <Contact> composite should be the *first* element in PR.19, immediately following any <ContentItem> composites and preceding any <ImprintName> or <Imprint> data.

Note the use of <Contact> in ONIX 2.1 is an alternative to the use of <PromotionContact>, and an ONIX 2.1 rev.04 record may contain both <Contact> and <PromotionContact>. The former need not be concerned solely with promotion of the product.

Experience with the use of <Contact> in ONIX 2.1 will inform how <Contact> might be carried forward into a future revision of ONIX 3 – for example it could become repeatable, with roles associated with multiple contacts. However, if <Contact> is carried forward into ONIX 3, the existing ONIX 3 <PromotionContact> is at market level and there may be different <PromotionContact> names for different markets. In contrast, <Contact> would remain at global ‘publisher’ level – there may well be a need for mechanisms to carry multiple role-based contacts at both levels.

1.6 Publication order of a series

Series titles are often published ‘out of order’ – that is, the order of publication is not necessarily the logical reading order or the numbered order of the complete series. For example, number 3 in a five-part series may be published first, followed by 1, 2, 4 then 5.

The new and optional data element <PubSequenceNumberWithinSeries> is proposed, to carry the actual publication order. This would immediately follow the element <NumberWithinSeries>, and would carry an integer:

```
<!-- for a book that is #3 in the series, but published first -->
<NumberWithinSeries>3</NumberWithinSeries>
<PubSequenceNumberWithinSeries>1</PubSequenceNumberWithinSeries>
```

1.7 Cut-off date for reservation orders

Reservation orders are handled via an addition to the <PriceQualifier>, PR.24.50. Since this is an important factor in the Japanese book supply chain, Appendix 1 illustrates the existing ONIX 2.1 method to handle special pre-publication pricing that is valid up until a specific date.

1.8 Sets where items are ‘Not for individual resale’

ONIX views a product that is a pack consisting of multiple sub-products (eg a pack containing a particular set of books) as an indivisible product, where the contents of the set cannot be separated and sold individually. No special notation needs to be added – *by default*, the items in the pack are considered to be not for individual resale.

Packs that are intended to be split, or may be split by the retailer, should be treated as trade products.

Appendix 2 illustrates how various types of pack may be described.

2 Additions to Codelists (included in Issue 12, October 2010, and in Issue 13, March 2011)

Various additions have already been made to support requirements of the Japanese book trade.

2.1 <MainSubject> and <Subject>

Extend lists 26 and 27 as follows (new in Codelists issue 12):

78	Japanese book trade C-Code
79	Japanese book trade Genre Code

2.2 <NameCodeType>

Extend list 44 as follows (new in Codelists issue 12):

19	Japanese Publisher identifier (administered by Japanese ISBN Agency)
----	--

2.3 <ProductFormDetail>

Extend lists 78 and 175 as follows, to add various standard Japanese formats (pre-existing codelist entries for typical Japanese sizes are shaded, new entries in Codelists Issue 12 are B119 to B129):

B108	A5 Tankobon	210 x 148mm HB
B109	B5 Tankobon	257 x 182mm HB
B110	B6 Tankobon	182 x 128mm HB
B111	A6 Bunko	148 x 105mm PB
B112	B40-dori Shinsho	182 x 103mm PB
B117	Mook	PB
B119	46 size	188 x 127mm
B120	46-Henkei size (near to 46 size)	
B121	A4 size	297 x 210mm
B122	A4-Henkei size (near to A4 size)	
B123	A5-Henkei size (near to A5 size)	
B124	B5-Henkei size (near to B5 size)	
B125	B6-Henkei size (near to B6 size)	
B126	AB size	257 x 210mm
B127	B7 size	128 x 91mm
B128	Kiku size	218 x 152mm
B129	Kiku-Henkei size (near to Kiku size)	

2.4 <PriceQualifier>

Extent List 59 to include a code for Reservation orders (new in Codelists issue 13):

07	Reservation order price – price valid for a specified period prior to publication. Orders places prior to the end of the period are guaranteed to be delivered to the retailer before the nominal publication dte. The price may or may not be different from the ‘normal’ price pertaining after expiry of the reservation order period, which carries no such delivery guarantee. Must be accompanied by a <PriceEffectiveUntil> date (or equivalent <PriceDate> composite in ONIX 3), and should also be accompanied by a ‘normal’ price with a <PriceEffectiveFrom> date.
----	---

3 Arrangements for validation of ONIX 2.1 rev.04 (JP) messages

ONIX XML data is frequently validated using a DTD hosted by EDItEUR on its website. This centrally-hosted DTD reflects version 2.1 rev.03 of the specification. However, since the release of rev.03, EDItEUR has moved away from hosting the DTDs and schemas for validation purposes. There is no centrally-hosted DTD or schema for ONIX 3.

It is proposed that Japanese organisations should make internal arrangements to validate their messages, by hosting a local copy of the DTD or schema within their own intranets. For transmission between organizations, a message should begin with the XML and !DOCTYPE declarations as normal, in order to comply with the standard:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXMessage SYSTEM
  "http://www.editeur.org/onix/2.1/reference/onix-international.
  dtd">
```

but for validation, the DTD should be drawn from a local source. Alternatively, validation should be performed using a local schema.

Note that any message that includes the new <Contact> composite, the new <PubSequenceNumberWithinSeries>, new *textscript* or *collationorder* attributes, or any XHTML that includes <ruby> markup will *not* validate against the centrally-hosted rev.03 DTD.

New DTD and schema files have been prepared for ONIX 2.1 rev.04, and will be supplied to the Japanese ONIX national group.

Appendices

Appendix 1 Cut-off date for reservation orders

The following snippet of an ONIX product record illustrates how a special price for orders placed prior to a specified date may be handled.

```
<Price>
  <PriceTypeCode>23</PriceTypeCode>
  <DiscountPercent>40</DiscountPercent>
  <PriceAmount>1900</PriceAmount>
  <CurrencyCode>JPY</CurrencyCode>
  <PriceEffectiveUntil>20100930</PriceEffectiveUntil>
</Price>
<Price>
  <PriceTypeCode>03</PriceTypeCode>
  <DiscountPercent>40</DiscountPercent>
  <PriceAmount>2100</PriceAmount>
  <CurrencyCode>JPY</CurrencyCode>
  <PriceEffectiveFrom>20101001</PriceEffectiveFrom>
</Price>
```

Price type 23 is 'Pre-publication fixed retail price excluding tax' and there are similar codes for recommended prices, and for prices that are inclusive of tax.

The above describes the case where the fixed or recommended retail price of the product itself changes (from ¥1900 to ¥2100 on 1st October in the example above). If the price to the consumer stays the same, but some special discount is given for early orders from retailers, the following would be used:

```
<Price>
  <PriceTypeCode>23</PriceTypeCode>
  <DiscountPercent>45</DiscountPercent>
  <PriceAmount>2100</PriceAmount>
  <CurrencyCode>JPY</CurrencyCode>
  <PriceEffectiveUntil>20100930</PriceEffectiveUntil>
</Price>
<Price>
  <PriceTypeCode>03</PriceTypeCode>
  <DiscountPercent>40</DiscountPercent>
  <PriceAmount>2100</PriceAmount>
  <CurrencyCode>JPY</CurrencyCode>
  <PriceEffectiveFrom>20101001</PriceEffectiveFrom>
</Price>
```

Here the advantageous discount terms offered to retailers expire (discount drops from 45% to 40% on 1st October). Of course, discounts may be coded rather than given as percentages, particularly if the exact discount is commercially sensitive.

For reservation orders, where orders placed up until a certain date are guaranteed to be fulfilled prior to publication, a <PriceQualifier> may be added:

```
<Price>
  <PriceTypeCode>23</PriceTypeCode>
  <PriceQualifier>07</PriceQualifier>
  <DiscountPercent>40</DiscountPercent>
  <PriceAmount>2100</PriceAmount>
  <CurrencyCode>JPY</CurrencyCode>
  <PriceEffectiveUntil>20100930</PriceEffectiveUntil>
</Price>
<Price>
  <PriceTypeCode>03</PriceTypeCode>
  <DiscountPercent>40</DiscountPercent>
  <PriceAmount>2100</PriceAmount>
```

```
<CurrencyCode>JPY</CurrencyCode>
<PriceEffectiveFrom>20101001</PriceEffectiveFrom>
</Price>
```

Note that, as in this example, neither the price nor the discount terms need vary – although they may vary if required. The only difference between the prices before 1st Oct and after 1st Oct is that the former comes with guaranteed delivery before the nominal publication date. <PriceEffectiveUntil> is the last date on which a reservation order may be placed.

Appendix 2 Sets where items are ‘Not for individual resale’

A pack may consist of (for example) four paperback books, shrinkwrapped together. The pack is described differently in ONIX, depending on whether the four books are identical or different, and on whether the four must be retailed together as a single product, or may be split and sold individually.

Four different books, may be separated by the retailer and sold individually	Product form	XL
	Product form additional	Four paperback books
	Product packaging	21
	Number of pieces †	4
	Contained items	Four repeats of composite, each with Item quantity = 1
	Each of the four books should also have an individual <Product> record (definitely so if they are available to the retailer individually, as well as in the pack)	
Four different books, may NOT be separated by the retailer and sold individually	Product form	BC
	Product form additional	Four paperback books
	Product packaging	21
	Number of pieces	4
	Contained items	Four repeats of composite, each with Item quantity = 1
	Each of the four books should also have an individual <Product> record, and may be Unpriced item type = 03 if they are not available individually at all	
Four identical books, may be separated by the retailer and sold individually	Product form	XL
	Product form additional	Four paperback books
	Product packaging	21
	Number of pieces	4
	Contained items	One repeat of composite, with Item quantity = 4
	Book should also have an individual <Product> record (definitely so if it is available to the retailer as single copies, as well as in the pack)	
Four identical books, may NOT be separated by the retailer and sold individually	Product form	WX
	Product form additional	Four paperback books
	Product packaging	21
	Number of pieces	4
	Contained items	One repeat of composite, with Item quantity = 4
	Book should also have an individual <Product> record, and may be Unpriced item type = 03 if it is not available individually at all	

† <NumberOfPieces> may occur either inside or outside the <ContainedItem> composite. In the table, Number of pieces refers to its use *outside* the composite