EDItEUR

ONIX Product Information **Guidelines Release 2.0** XML message specification

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

These *Guidelines* were compiled for EDItEUR by David Martin.   The XML DTD which defines the communication format for ONIX product information messages was developed by Francis Cave, who also contributed much of the content of this document.

TERMS AND CONDITIONS OF USE OF THE ONIX PRODUCT INFORMATION XML DTD

All ONIX standards and documentation are copyright materials, made available free of charge for general use. If you use the ONIX Product Information DTD, you will be deemed to have accepted these terms and conditions:

1. You agree that you will not add to, delete from, amend, or copy for use outside of the ONIX Product Information DTD, any part of the DTD except in those ways which are expressly described in ONIX documentation.

2. You agree that if you wish to use the XML supergroup "Z" which is provided for local extensions, you will in the first instance notify EDItEUR and allow EDItEUR to review and comment on your proposed use.

3. You further agree that if, in the light of EDItEUR comments, you proceed to develop an application of the XML supergroup "Z", you will provide EDItEUR with a copy of your extended ONIX Product Information DTD, including any supporting documentation that is required to understand fully the nature and application of the extension, and EDItEUR will be free to make such use as it sees fit of any part of your application for the future development of ONIX.

If you do not accept these terms, you must not use the ONIX Product Information DTD.

EDItEUR is the international group which coordinates the development and promotion of standards for electronic commerce in the book and serials sectors.

EDItEUR
c/o Book Industry Communication
39-41 North Road
LONDON N7 9DP
UK

Telephone +44 (0)20 7607 0021
Fax +44 (0)20 7607 0415
http://www.editeur.org/

Book Industry Study Group, Inc (BISG)
750 Route 34
Suite 1
Matawan NJ 07747
USA

Telephone +1 (732) 583 0066
Fax +1 (732) 583 3652
http://www.bisg.org/

Book Industry Communication (BIC)
39-41 North Road
LONDON N7 9DP
UK

Telephone +44 (0)20 7607 0021
Fax +44 (0)20 7607 0415
http://www.bic.org.uk/

Contents

If you have comments, questions or suggestions for improvements to these *Guidelines*, please send them to one of the following contact points:

For the USA: Frank Daly, BISG, email *frank@bookinfo.org*

For the UK and all other countries: Brian Green, EDItEUR, email *brian@bic.org.uk*

1. Release 2.0 notes

In Release 2.0 the XML message specification has for the first time been separated from the content *Guidelines*. Apart from the updating of the release number throughout, the following substantive changes will be found in this new document:

CHANGES IN RELEASE 2.0

Page 5	A new sender contact email address element MH.5 has been added, to allow an email address for queries to be sent as a separate field rather than as part of field MH.4.
Page 8	The short tag for the default unit of weight has been corrected to <m188>.
Page 10	Section 4 is entirely new, and describes the set of XML attributes which have been added in Release 2.0.
Page 14	Corrections have been made to the description of the basic ASCII character set supported in XML.
Page 16	The advice on using HTML in ONIX text fields has been revised to show the use of the new "textformat" attribute.

CORRECTIONS, 8 August 2001

Page 5	A cut-and-paste error affecting the definitions for the new element MH.5 has been corrected.
Page 7	The example of a default linear unit in MH.17 has been corrected.
Page 9	In the short tag example, tags around “David Martin” have been corrected.
Page 14	Section 6.1: the third currency symbol listed here is the “Euro”, which may not print or display for everybody, so an explanatory note has been added.

2. ONIX XML product information message

An ONIX product information message can be regarded as having four component parts: the start of message, whose format and content is dictated by the XML standard; a message header block, which is specific to ONIX; the body of the message; and the end of message.

2.1 Start of message

The start of message consists of a number of so-called "declarations".

In the simplest case, if you are using the ONIX "short tags", an ONIX product information message compliant with Release 2.0 should always begin with these four lines:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.0/short/onix-international.dtd">
<ONIXmessage>
```

The first line, the "XML declaration", simply identifies this as a document which is defined in accordance with the XML standard, version 1.0.

The second and third lines together constitute the "DTD declaration", which tells standard XML software that this is a document which has been constructed in accordance with Release 2.0 of the ONIX XML DTD using short tag names. (Note that the second and third lines may be run together, with a space at the join, but splitting them over two lines is considered to be more readable.)

The URI specified in the third line provides an unambiguous, active reference to the short tag name version of Release 2.0 of the ONIX DTD. This URI links to a location on the EDItEUR website where the files corresponding to the short tag name version of Release 2.0 DTD are held. Any subsequent releases of, and any intermediate revisions to, the ONIX DTD will need to be referred to by distinct URIs. See Section 5 for more detailed information on the identification of different releases of, and revisions to, the ONIX DTD.

The fourth line says that an ONIX product information message starts here.

If you are using the longer "Reference Names", or if you want to define your own data element labels, see Section 5.

2.2 Message header block

After these opening lines, the message header carries a number of data elements, specifying the sender and date of message (mandatory), the addressee (optional), and optionally stating message default values for measure units, currency etc.

Sender EAN number
Sender SAN
Sender company name
Sender contact person
Sender email address
Addressee EAN number
Addressee SAN
Addressee company name
Addressee contact
Message sequence number
Message repeat number
Message creation date/time
Message note
Default language of text
Default price type
Default currency
Default unit of linear measurement
Default unit of weight
Default class of trade

These are specified in detail in section 3.

The message header elements are "packaged" into a **<Header>** composite, to make it easy for systems which repeatedly use essentially the same header content to reference it as a unit. It is not mandatory to send them as a composite – the DTD will still allow the elements to be sent as in previous releases, and this remains the Level 1 convention.

2.3 Body of message

The body of an ONIX product information message consists of one or more Product, Main Series, and/or Subseries records, in any mix, and with no theoretical limit on the number of records. Each of these three record types is specified in a separate ONIX product information *Guidelines* document.

Note that for most applications the Main Series and Subseries records are not expected to be required, since series detail can be, and normally is, carried within a Product record.

2.4 End of message

The end of message "trailer" is a single line:

```
</ONIXmessage>
```

2.5 Layout of a complete message

In summary, the layout of a typical ONIX product information message is like this:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.0/short/onix-international.dtd">
<ONIXmessage>
      <header>
            Message header data elements
      </header>
      <product>
            Product information data elements for product 1
      </product>
      <product>
            Product information data elements for product 2
      </product>
      <product>
            .......................
            .......................
      </product>
</ONIXmessage>
```

2.6 Using the ONIX XML DTD

Your attention is drawn to the terms and conditions of use which appear in the DTD itself and on the reverse of the cover page of these *Guidelines*.

3. Message header content

Header composite

A group of data elements which together constitute a message header. The elements may alternatively be sent without being grouped into a composite, but the composite approach is recommended since it makes it easier to maintain a standard header "package" to drop into any new ONIX Product Information message.

Reference name	<Header>
Short tag	<header>

MH.1 Sender EAN number

An EAN location number which identifies the sender of an ONIX message. Optional and non-repeating.

Format	Fixed-length, thirteen numeric digits, of which the last is a check digit.
Reference name	<FromEANNumber>
Short tag	<m172>
Example	*5401234098123*

MH.2 Sender SAN **USA only**

A US book trade Standard Address Number which identifies the sender of an ONIX message. Optional and non-repeating.

Format	Fixed-length, seven characters. The first six are numeric digits, and the seventh is a check character which may be a numeric digit or letter X.
Reference name	<FromSAN>
Short tag	<m173>
Example	*978847X*

MH.3 Sender company name

The name of the sender organization, which should always be stated in a standard form agreed with the addressee. Mandatory and non-repeating.

Format	Variable-length ASCII text, suggested maximum 30 characters
Reference name	<FromCompany>
Short tag	<m174>
Example	*HarperCollins London*

MH.4 Sender contact

Free text giving the name, department, phone number, etc for a contact person in the sender organization who is responsible for the content of the message. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 300 characters
Reference name	<FromPerson>
Short tag	<m175>
Example	*Jackie Brown, 020 7979 6444*

MH.5 Sender contact email address

A text field giving the email address for a contact person in the sender organization who is responsible for the content of the message. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 100 characters
Reference name	<FromEmail>
Short tag	<m283>
Example	*jackie.brown@bigpublisher.co.uk*

MH.6 Addressee EAN number

An EAN location number which identifies the addressee of an ONIX message. Optional and non-repeating.

Format	Fixed-length, thirteen numeric digits, of which the last is a check digit.
Reference name	<ToEANNumber>
Short tag	<m176>
Example	*5401234098123*

MH.7 Addressee SAN USA only

A US book trade Standard Address Number which identifies the addressee of an ONIX message. Optional and non-repeating.

Format	Fixed-length, seven characters. The first six are numeric digits, and the seventh is a check character which may be a numeric digit or letter X.
Reference name	<ToSAN>
Short tag	<m177>
Example	*978847X*

MH.8 Addressee company name

The name of the addressee organization, which should always be stated in a standard form agreed with the addressee. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 30 characters
Reference name	<ToCompany>
Short tag	<m178>
Example	*Book Data*

MH.9 Addressee contact

Free text giving the name, department etc for a contact person in the addressee organization to whom the message is to be directed. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 300 characters
Reference name	<ToPerson>
Short tag	<m179>
Example	*Mel Carter*

MH.10 Message sequence number

A sequence number of the messages in a series sent between trading partners, to enable the receiver to check against gaps and duplicates. Optional and non-repeating.

Format	Variable-length integer,
Reference name	<MessageNumber>
Short tag	<m180>
Example	*1234*

MH.11 Message repeat number

A number which distinguishes any repeat transmissions of a message. If this element is used, the original is numbered 1 and repeats are numbered 2, 3 etc. Optional and non-repeating.

Format	Variable-length integer
Reference name	<MessageRepeat>
Short tag	<m181>
Example	*2*

MH.12 Message creation date/time

The date on which the message is sent. Optionally, the time may be added, using the 24-hour clock. Mandatory and non-repeating.

Format	Eight or twelve numeric digits only (YYYYMMDD or YYYYMMDDHHMM)
Reference name	<SentDate>
Short tag	<m182>
Example	*200005220230*

MH.13 Message note

Free text giving additional information about the message. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 500 characters
Reference name	<MessageNote>
Short tag	<m183>
Example	*Updates for titles to be published September 2000*

MH.14 Default language of text

An ISO standard code indicating the default language which is assumed for the text of products listed in the message, unless explicitly stated otherwise by sending a “language of text” element in the product record. Optional and non-repeating.

Format	Fixed-length, three letters.
Code list	As specified for field PR.11.1 in the product record
Reference name	<DefaultLanguageOfText>
Short tag	<m184>
Example	*eng*

MH.15 Default price type

An EPICS code indicating the default price type which is assumed for prices listed in the message, unless explicitly stated otherwise in a **<Price>** composite in the product record. Optional and non-repeating.

Format	Fixed-length, two numeric digits.
Code list	As specified for field PR.24.28 in the product record
Reference name	<DefaultPriceTypeCode>
Short tag	<m185>
Example	*01*

MH.16 Default currency

An ISO standard code indicating the currency which is assumed for prices listed in the message, unless explicitly stated otherwise in a **<Price>** composite in the product record. Optional and non-repeating.

Format	Fixed-length, three letters.
Code list	As specified for field PR.24.40 in the product record
Reference name	<DefaultCurrencyCode>
Short tag	<m186>
Example	*USD*

MH.17 Default linear unit

A code indicating the default unit which is assumed for linear measurements listed in the message, unless otherwise specified in the product record. **This element is provided for use at Level 1. For most implementations, explicit coding of measure units with each occurrence of a measurement is to be preferred.** Optional and non-repeating.

Format	Fixed-length, two letters.
Code list	As specified for field PR.22.3 in the product record (linear units only)
Reference name	<DefaultLinearUnit>
Short tag	<m187>
Example	*mm*

MH.18 Default unit of weight

A code indicating the default unit which is assumed for weights listed in the message, unless otherwise specified in the product record. **This element is provided for use at Level 1. For most implementations, explicit coding of units with each occurrence of a weight is to be preferred.** Optional and non-repeating.

Format	Fixed-length, two letters.
Code list	As specified for field PR.22.3 in the product record (units of weight only)
Reference name	<DefaultWeightUnit>
Short tag	<m188>
Example	*oz*

MH.19 Default class of trade USA only

Free text indicating the class of trade which is assumed for prices given in the message, unless explicitly stated otherwise specified in **<j149>**. For example: *Institutional, General trade, Wholesale distributor*, which may be represented by a suitable code or abbreviation agreed between trading partners. otherwise specified in the product record. Optional and non-repeating.

Format	ASCII text, suggested maximum length 50 characters.
Reference name	<DefaultClassOfTrade>
Short tag	<m193>
Example	*gen*

End of header composite

Example of an ONIX product information message header

Using reference names

```
<?xml version="1.0"?>
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.0/reference/onix-international.dtd">
<ONIXmessage>
<header>
      <FromCompany>Portadas.net</FromCompany>
      <FromPerson>Bernie Rabow bernie.rabow@portadas.net</FromPerson>
      <ToCompany>EDItEUR</ToCompany>
      <ToPerson>David Martin</ToPerson>
      <MessageNumber>1213</MessageNumber>
      <SentDate>200007311330</SentDate>
      <MessageNote>Prueba en Espa&ntilde;ol</MessageNote>
      <DefaultLanguageOfText>spa</DefaultLanguageOfText>
</header>
```

Using short tags

```
<?xml version="1.0"?>
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.0/short/onix-international.dtd">
<ONIXmessage>
<header>
      <m174>Portadas.net</m174>
      <m175>Bernie Rabow bernie.rabow@portadas.net</m175>
      <m178>EDItEUR</m178>
      <m179>David Martin</m179>
      <m180>1213</m180>
      <m182>200007311330</m182>
      <m183>Prueba en Espa&ntilde;ol</m183>
      <m184>spa</m184>
</header>
```

4. Use of XML attributes

In Release 2.0, we are using XML attributes for the first time. The view which has been taken in the development of ONIX is that it is undesirable to use XML attributes to carry portions of the actual data content of the ONIX message. However, it is appropriate to use them to carry information which qualifies the data itself and its representation – metadata about metadata, as it were.

Accordingly, six attributes are now defined in the ONIX Product Information DTD, as follows:

Text format	
Function	Enables the format of any text element to be specified in the same way as was previously only possible in the **<OtherText>** composite. Intended in due course to replace the <TextFormat> element in that composite.
Form	`textformat=”code”`
Code list (taken from the <TextFormat> field PR.15.4)	00 ASCII text (default) 01 SGML 02 HTML 03 XML

Language	
Function	Enables the language of any text element to be specified when it is not the expected default language of the message.
Form	`language=”code”`
Code list	ISO 639-2/B language codes (see Product record, PR.11.1)

Transliteration	
Function	Enables a transliteration scheme used in a text element to be specified.
Form	`transliteration=”code”`
Code list	To be defined

Datestamp	
Function	Enables any data element or composite to carry the date or date and time when it was last changed.
Form	`datestamp=”YYYYMMDD” or ”YYYYMMDDHHMMSS”`

Source type	
Function	Enables a data element or composite to carry a code indicating the type of source or authority for the data content.
Form	**sourcetype="**`code`**"**
Code list (as in the <RecordSourceType> field PR.1.5)	00 Unspecified (default value) 01 Publisher 02 Publisher's distributor (use only for a distributor appointed by the publisher, as distinct from a wholesaler) 03 Wholesaler 04 Bibliographic agency 05 Library bookseller

Source name	
Function	Enables a data element or composite to carry the name of the source or authority for the data content.
Form	**sourcename="**`name`**"**

Note that the XML DTD will allow any of these attributes to be carried in any element. However, some of them are plainly meaningless except when applied to a text data element, or to a composite which includes text data elements.

Attributes are carried within an XML tag, for example **`<Text textformat="02">`**. The attribute name is separated from the name of the element by a space, and the attribute value is placed in quotes. If there are two or more attributes in a single tag, they too are separated by a space, for example **`<Text textformat="02" language="eng">`**.

It is not expected that this new facility will find substantial immediate take-up, except that it is strongly recommended that the "textformat" attribute should now be preferred to the <TextFormat> field in the Product record, PR.15.4.

Technically-minded XML users will be aware that, at least in the case of "textformat" and "sourcetype", we could be using the DTD to constrain the values of these attributes to be taken from the specified code sets. The ONIX development team has taken the view that it would be wrong to introduce such constraints into the DTD where they could only be applied to two or three attributes. Given that there is no mechanism in the DTD to constrain data element content to conform to the code sets, value ranges and string lengths defined in the guidance documentation, to make an exception of these attributes would be inconsistent. A generalised mechanism for controlling both attribute values and data element content would be provided by an enhanced specification using a schema definition language rather than the DTD, and it is likely that future releases of ONIX will be supported by full schemas.

For the same reason as above, the DTD does not define default values for any of the new attributes, even though default values are given in the above table.

5. Start of message "declarations"

At the start of an ONIX product information message, there is a series of XML "declarations": the *XML declaration*, the *DTD declaration*, and the *namespace declaration*, which is part of the message start tag.

In this section we give more details of each of these, and show how they are used to define various options for ONIX users.

5.1 XML declaration

The *XML declaration* simply states that your message is in XML. It should contain the following sequence of characters:

```
<?xml version="1.0"?>
```

If you intend to use a character set other than ASCII for your message, you will need to use a modified form of XML declaration as detailed below in section 6.

5.2 DTD declaration

The *DTD declaration* (or *document type declaration*, to be more precise) identifies the XML DTD against which your message should be validated.

Given that the ONIX product information DTD will be subject to revision by EDItEUR from time to time, and may also be modified to meet local requirements, it is essential that the precise version of the DTD used in creating the message, and therefore to be used subsequently in validating the message, should be clearly and unambiguously identified. The master versions of the ONIX DTD are held by EDItEUR, and for this reason the DTD should normally be identified by the URI where it is to be found on the EDItEUR website. If short tag names are used, the URI for Release 2.0 is:

```
http://www.editeur.org/onix/2.0/short/onix-international.dtd
```

If the reference tag names are used, the URI for Release 2.0 is:

```
http://www.editeur.org/onix/2.0/reference/onix-international.dtd
```

If the DTD is modified to meet local needs in a way approved by EDItEUR (normally this would only involve changes to tag names – see separate guidance on defining local extensions to the ONIX DTD), this URI should be replaced by one that locates the modified version, enabling recipients of messages that use it to retrieve it for validation purposes.

Future major releases of the ONIX DTD by EDItEUR will be identified by changes to the release number, which will be reflected in a corresponding change to the URI, eg "2.0" being replaced by "2.1" in the above URI.

In the event of intermediate minor revisions being made to the DTD between major releases, these revisions will in normal circumstances replace any previous revisions to the current release. If, for any reason, a minor revision other than the current revision needs to be identified as being used for a specific message, the URI should be modified to include reference to the revision number, eg

```
http://www.editeur.org/onix/2.0/nn/short/onix-international.dtd
```

where "nn" is a 2-digit revision number commencing with "01" for the first revised version of this release.

The complete standard DTD declaration for the short tag name version of Release 2.0 of the DTD should contain the following sequence of characters:

```
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.0/short/onix-international.dtd">
```

The complete standard DTD declaration for the reference tag name version of Release 2.0 of the DTD should contain the following sequence of characters:

```
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.0/reference/onix-international.dtd">
```

If you are using a local set of variant names, the DTD declaration will need to be modified to refer to your modified version of "*onix-international.dtd*", which contains the variant names. If this modified file is available on a web site, your DTD declaration should read:

<!DOCTYPE *MyMessageTagName* **SYSTEM "***URI***">**

where "*MyMessageTagName*" is your local equivalent of the standard message tag name "**ONIXmessage**", and "*URI*" is a Uniform Resource Identifier for the modified version of "*onix-international.dtd*".

If you are using local variant names, but don't plan to make them available on a web server, you will need to issue copies of the modified version of "*onix-international.dtd*" to all recipients of your messages. In this case the URI may be replaced simply by the filename of the modified file, and the recipient will ensure that their system can find it.

Note – The DTD comprises five files, but only the one referred to above as "*onix-international.dtd*" is referred to directly by the DTD declaration. This file contains a reference to a second file, which in its turn contains references to the other three files.

5.3 Namespace declaration

If you are using the short tag names, the message start tag should contain the following sequence of characters:

```
<ONIXmessage>
```

If you are using the reference (verbose) names, the message start tag may contain the following sequence of characters (note the use of a capital "M" in the reference tag name "**ONIXMessage**"), but this is not obligatory:

```
<ONIXMessage xmlns="http://www.editeur.org/onix/ReferenceNames/">
```

If you are using a set of local variant names, the message start tag should contain the following sequence of characters:

<*MyMessageTagName* **xmlns="***URI***">**

where "*MyMessageTagName*" is your local tag name in place of "**ONIXmessage**", and "*URI*" is a Uniform Resource Identifier that uniquely identifies the names you are using, eg:

```
<MyMessageTagName
xmlns="http://www.mycompany.com/UseOurONIXTagNames/">
```

Please note that the URI does NOT have to correspond to an actual Web address that is reachable by a browser. It is simply a device for creating an unambiguous reference to the authority for a defined namespace.

For further information on namespaces see http://www.w3.org/TR/REC-xml-names.

6. Character sets and special characters

6.1 Using special characters and extended character sets in ONIX messages

The character set of XML messages is assumed to be the printable characters of ASCII (ie those characters whose numbers fall between 32 and 126 inclusive), unless the XML declaration includes an “encoding declaration” which specifies the character set of the message. For reference here is a list of the basic character set for which **no** special coding is necessary, and all of which can be found on a standard computer keyboard:

Space character
Capital letters: **A – Z**
Lower-case letters: **a – z**
Digits: **0 – 9**
Punctuation and brackets: **!”’(),-.:;?[]{}**
Currency, arithmetic, computer and other symbols: **#$%*+/=>\@_`|~**

This set deliberately does **not** include the characters “&” and “<”. These characters have special significance in all XML applications and should never be included in the text of an XML message in any way other than is described in section 6.2 below.

Note that the currency symbols “**£**”, “**¥**” and “ ” (the “Euro” symbol, which for some readers may not print or display when they view this page) are **not** in the basic set. Note also that some office applications insert so-called “smart” left- and right-handed quotation marks (single or double), which are **not** in the basic set. En and Em rules are **not** in the basic set.

If an extended character set is to be used, the modified XML declaration should take the form:

<?xml version="1.0" encoding=”*EncName*”?>

where *EncName* is the name of a character set encoding which conforms with the requirements of Section 4.3.3 of the *XML 1.0 Recommendation*.

The default character set encodings of the *XML 1.0 Recommendation* are the Unicode sets UTF-8 (of which ASCII is a subset) and UTF-16. Where the use of special character entities is not a satisfactory option, it is recommended that UTF-8 or UTF-16 should be used to extend the ASCII range. Implementers should note that it is NOT a requirement that recipients of ONIX messages should be able to handle correctly any character set encodings other than ASCII. Local circumstances may favour the use of other character set encodings, such as any of Parts 1 through 9 of ISO 8859 and various Asian language encodings. For further guidance on character encodings in XML see Section 4.3.3. of the *XML 1.0 Recommendation*.

However, all releases of the ONIX Product Information DTD from Release 1.2 onwards include the three special character entity sets contained in XHTML 1.0, the XML-compliant form of HTML (see http://www.w3.org/TR/xhtml1/), ie:

Latin-1 characters (xhtml-lat1.ent)
Special characters (xhtml-special.ent)
Symbols (xhtml-symbol.ent)

These provide a basis for encoding as XML expressions a number of the most commonly-used special characters which are not part of the ASCII set. The form of expression used is the XML entity reference, defined in Section 4.1 of the *XML 1.0 Recommendation*. For example, a lower-case e with acute accent “é” may be encoded:

```
&eacute;
```

Experienced XML users will be aware that XML character references are a form of XML expression that gives access to the full Unicode character set. While this form of expression must be permissible in all XML applications, its use is not recommended here. Recipients of ONIX messages may well not have ready access to the full Unicode character set and may therefore have difficulty in interpreting references to arbitrary Unicode characters.

6.2 Coding the ampersand "&" and less-than symbol "<"

The characters "&" and "<" have special meaning in XML, and therefore cannot be used as text characters in ONIX data elements. If you need to include either of these characters in a data element (for example in the name of an organization such as "AT&T" that by convention uses "&" rather than "and") you will need to include the XML entity reference form of expression in place of the "&" or "<" as specified in the following table:

Desired character	XML expression
&	**`&amp;`**
<	**`&lt;`**

In the above example, "AT&T" should therefore be expressed in an ONIX message by the sequence

```
AT&amp;T
```

7. Using HTML in ONIX text fields

HTML tags may be used with text in appropriate ONIX data elements (eg in product descriptions and other supporting text). The use of HTML tags in any data element should be signalled by specifying the value of the “textformat” attribute (see section 4) in the start tag of the data element in question, eg:

```
<MainDescription textformat=”02”>... HTML-tagged text ...
...</MainDescription>
```

In the specific case of the composite element <OtherText> in Group 18, the data element <TextFormat> can be used to signal that the following <Text> data element contains HTML-tagged text. However, the availability of the “textformat” attribute on all data elements, including <Text>, makes the use of <TextFormat> unnecessary, and it is therefore discouraged.

Two methods are available for including HTML tags in the text of an ONIX data element, and have been described in previous releases of these *Guidelines*:

Method 1. The entire text of the data element is enclosed within an XML “CDATA Section” (see Section 2.7 of the *XML 1.0 Recommendation*). For example:

```
<Text textformat=”02”><![CDATA[<p>HTML-tagged review quotation... <i>text
in italics</i> ...</p>]]></Text>
```

A conforming XML system will not recognise any XML markup tags inside a CDATA Section. In particular, it will not recognise special character entity references. Any HTML-tagged text included in a CDATA Section must therefore be extracted from the ONIX message before it can be checked for tagging errors.

Because of this limitation, Method 1 is *not* recommended.

Method 2. The “**<**” character at the start of every HTML element start and end tag is replaced by its entity reference as described in the previous section. For example:

```
<Text textformat=”02”>&lt;p>HTML-tagged review quotation... &lt;i>text in
italics&lt;/i> ...&lt;/p></Text>
```

The advantage with this second method is that a conforming XML system will recognise any special character entity references in the HTML-tagged text.

Method 2 is the *recommended* method for including HTML

NOTE – The technique of replacing the “**<**” symbol by “**&lt**;” is often referred to as “escaping” the associated tag. While in principle it is also possible to “escape” special character entity references in the same manner (eg “**&eacute;**” replaced by “**&amp;eacute;**”), in practice this is considered to be neither necessary nor desirable, since in normal conditions it prevents these characters from being displayed correctly. Escaping entity references is therefore **not** recommended.