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

ONIX for Books
Product Information Message
XML Message Specification

Release 2.1, revision 03 January 2006

This XML Message Specification was compiled for EDItEUR by David Martin and Francis Cave. The XML DTD and XML Schema which define the communication format for *ONIX for Books* Product Information Messages were developed by Francis Cave.

Copyright © 2006 EDItEUR Limited. All rights reserved.

TERMS AND CONDITIONS OF USE OF THE *ONIX for Books* PRODUCT INFORMATION MESSAGE XML DTD AND XML SCHEMA

All ONIX standards and documentation are copyright materials, made available free of charge for general use. If you use the *ONIX for Books* Product Information Message DTD or XML Schema, 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 for Books* Product Information Message DTD or XML Schema, any part thereof except for strictly internal use in your own organisation.

2. You agree that if you wish to add to, amend, or make extracts of the DTD or XML Schema for any purpose that is not strictly internal to your own organisation, you will in the first instance notify EDItEUR and allow EDItEUR to review and comment on your proposed use, in the interest of securing an orderly development of the DTD or XML Schema for the benefit of other users.

If you do not accept these terms, you must not use the *ONIX for Books* Product Information Message DTD or XML Schema.

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 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/

Book Industry Study Group, Inc (BISG)
19 West 21st Street - Suite 905
New York, NY 10010
USA

Telephone +1 (646) 336 7141
http://www.bisg.org/

Contents

Page

1. Release 2.1 notes 4
2. *ONIX for Books* XML Product Information Message 6
   2.1 Start of message 6
   2.2 Message header block 6
   2.3 Body of message 7
   2.4 End of message 7
   2.5 Layout of a complete message 7
   2.6 Using the ONIX DTD and XML Schema 8
3. Message header content 9
4. Use of XML attributes 17
5. Start of message "declarations" 19
   5.1 XML declaration 19
   5.2 DTD declaration 19
   5.3 Namespace declaration 20
   5.4 Start and end tags 21
   5.5 XML Schema references 21
6. Character sets and special characters 22
   6.1 Using special characters and extended character sets in ONIX messages 22
   6.2 Coding the ampersand "&" and less-than symbol "<" 23
7. Using XHTML, HTML, SGML or XML in ONIX text fields 24
   7.1 Introduction 24
   7.2 XHTML (Version 1.0 or later) 24
   7.3 HTML (Version 4.0 or earlier) 25
   7.4 Other SGML or XML 25

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: BISG, email *info@bookinfo.org*

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

1. Release 2.1 notes

Apart from the updating of the release number to 2.1 throughout, the following substantive changes will be found in this document:

CHANGES IN REVISION 03, January 2006

Section 5.3 On page 21, errors in the statement of namespace URIs have been corrected.

Section 7 On page 24, a new introductory section 7.1 has been added to provide background explanation on HTML and XHTML, and subsequent sections have been renumbered.

REVISIONS TO DOCUMENTATION, February 2005

Throughout References to the *ONIX for Books* XML Schema, which is now maintained in parallel with the DTD, have been added wherever appropriate.

Section 3 Code values for coded attributes have been deleted and replaced by links to separate HTML code lists.

CHANGES IN REVISION 02

Section 3 Notes on elements MH.1 to MH.6 have been revised to make it clear that, at minimum, *either* a sender identifier *or* a sender name must be included in the message header – the sender name *per se* is not mandatory, though it is normally expected.

Section 5 On page 20, a longstanding error in the DTD declaration for the reference name form of an ONIX message has been corrected.

On page 21, a new section 5.4 has been added to further clarify the differences between reference name and short tag forms.

Section 6 "XHTML 1.0 Latin-1 characters" has been deleted from the list of special character sets, to be consistent with the DTD. Related text has been revised.

CHANGES IN REVISION 01

Section 5 The use of namespace declarations with ONIX messages has been clarified. Namespace declarations should *not* be used in regular ONIX applications in which the ONIX DTD is being used for record or message validation purposes. They are, however, applicable when using ONIX tags in other XML messaging applications.

A more detailed description of the way in which successive minor revisions of the DTD are handled on the EDItEUR website has been added.

CHANGES IN RELEASE 2.1

Prelims The terms and conditions for use of the ONIX Product Information Message DTD have been simplified to reflect the removal of "supergroup" Z.

Section 3 New **<SenderIdentifier>** and **<AddresseeIdentifier>** composites have been added (within the **<Header>** composite only) for consistency with the ONIX policy of enabling identifiers to be used in a generalised way in any context where an organisation is named.

Hyperlinks to HTML code lists have been added in the same way as in other ONIX documentation.

Section 4	A new value 04 XHTML has been added to the code list for the **textformat** attribute. A new attribute **textcase** has been added.
Section 5	The text has been revised for clarification, without substantive change.
Section 6	The range of special character entities included in the DTD has been greatly enlarged.
Section 7	This section is substantially revised to describe a new facility for including XHTML in specified text elements. This is now the preferred way to send HTML in ONIX.

2. *ONIX for Books* XML Product Information Message

An *ONIX for Books* 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.

From Release 2.1 Revision 02 onwards, EDItEUR is maintaining formal definitions of the *ONIX for Books* Product Information Message both in DTD and XML Schema format. The emphasis in these pages is on the use of the DTD, since this is the more widely used. Supplementary notes on the XML Schema definition are included wherever appropriate.

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 “reference names”, an *ONIX for Books* Product Information Message compliant with Release 2.1 should always begin with these four lines:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXMessage SYSTEM
"http://www.editeur.org/onix/2.1/reference/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 2.1 of the *ONIX for Books* XML DTD (the “ONIX DTD”) using reference 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 reference name version of Release 2.1 of the ONIX DTD. This URI links to a location on the EDItEUR website where the files corresponding to the current reference name version of the Release 2.1 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 for Books* Product Information Message starts here.

If you are using the “short tags”, 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 identifier composite
Sender company name
Sender contact person
Sender email address
Addressee EAN number
Addressee SAN
Addressee identifier composite
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 ONIX DTD will still allow the elements to be sent as in previous releases.

2.3 Body of message

The body of an *ONIX for Books* 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 for Books* Product Information *Record Format* 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 for Books* Product Information Message, using reference names, is like this:

```
<?xml version="1.0"?>
<!DOCTYPE ONIXMessage SYSTEM
"http://www.editeur.org/onix/2.1/reference/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 DTD and XML Schema

Your attention is drawn to the terms and conditions of use which appear in the ONIX DTD and in the XML Schema themselves, and on the reverse of the cover page of these *Guidelines*.

Users of the *ONIX for Books* XML Schema are advised that, while the XML Schema is fully supported and maintained by EDItEUR, it is not a requirement that ONIX users generally – and recipients in particular – must support its use. If there is any doubt as to whether or not a particular recipient supports use of the XML Schema for ONIX, it must be assumed that they do not, and the ONIX message should contain the appropriate DTD Declaration as indicated in Section 2.1 above.

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. **Note that the Sender and Addressee Identifier composites can *only* be used within the Header composite, and future extensions to the Header will be defined *only* within the composite.**

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; but *either* the **<FromCompany>** element *or* a sender identifier using one or more elements from MH.1 to MH.5 *must* be included.

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

A US book trade Standard Address Number which identifies the sender of an ONIX message. Optional and non-repeating; but *either* the **<FromCompany>** element *or* a sender identifier using one or more elements from MH.1 to MH.5 *must* be included.

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*

Sender identifier composite

A group of data elements which together define the identifier of the sender within a specified namespace, used here to allow different party identifiers to be included without defining additional data elements. In particular the composite allows a proprietary identifier to be used by mutual agreement between parties to an exchange. The composite is optional and repeatable; but *either* the **<FromCompany>** element *or* a sender identifier using one or more elements from MH.1 to MH.5 *must* be included.

Reference name	<SenderIdentifier>
Short tag	<senderidentifier>

MH.3 Sender identifier type

An ONIX code identifying the namespace from which the identifier in the **<IDValue>** element is taken. Mandatory in any occurrence of the **<SenderIdentifier>** composite, and non-repeating.

Format	Fixed-length, two numeric digits
Code list	List 44
Reference name	<SenderIDType>
Short tag	<m379>
Example	*01*

MH.4 Identifier type name

A name which identifies a proprietary identifier type when, and only when, the code in the **<SenderIDType>** element indicates a proprietary scheme. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 50 characters
Reference name	<IDTypeName>
Short tag	<b233>
Example	

MH.5 Identifier value

An identifier of the type specified in the **<SenderIDType>** element. Mandatory in any occurrence of the **<SenderIdentifier>** composite, and non-repeating.

Format	According to the identifier type specified in **<SenderIDType>**
Reference name	<IDValue>
Short tag	<b244>
Example	

End of sender identifier composite

MH.6 Sender company name

The name of the sender organization, which should always be stated in a standard form agreed with the addressee. Optional and non-repeating; but *either* the **<FromCompany>** element *or* a sender identifier using one or more elements from MH.1 to MH.5 *must* be included.

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

MH.7 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.8 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.9 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.10 Addressee SAN

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*

Addressee identifier composite

A group of data elements which together define the identifier of the addressee within a specified namespace, used here to allow different party identifiers to be included without defining additional data elements. In particular the composite allows a proprietary identifier to be used by mutual agreement between parties to an exchange. The composite is optional and repeatable.

Reference name	<AddresseeIdentifier>
Short tag	<addresseeidentifier>

MH.11 Addressee identifier type

An ONIX code identifying the namespace from which the identifier in the **<IDValue>** element is taken. Mandatory in any occurrence of the **<AddresseeIdentifier>** composite, and non-repeating.

Format	Fixed-length, two numeric digits
Code list	List 44
Reference name	<AddresseeIDType>
Short tag	<m380>
Example	*01*

MH.12 Identifier type name

A name which identifies a proprietary identifier type when, and only when, the code in the **<AddresseeIDType>** element indicates a proprietary scheme. Optional and non-repeating.

Format	Variable-length ASCII text, suggested maximum 50 characters
Reference name	<IDTypeName>
Short tag	<b233>
Example	

MH.13 Identifier value

An identifier of the type specified in the **<AddresseeIDType>** element. Mandatory in any occurrence of the **<AddresseeIdentifier>** composite, and non-repeating.

Format	According to the identifier type specified in **<AddresseeIDType>**
Reference name	<IDValue>
Short tag	<b244>
Example	

End of addressee identifier composite

MH.14 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.15 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.16 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.17 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.18 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.19 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.20 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. This default will be assumed for all product records which do not specify a language in Group PR.11. Optional and non-repeating.

Format	Fixed-length, three letters.
Code list	List 74
Reference name	<DefaultLanguageOfText>
Short tag	<m184>
Example	*eng*

MH.21 Default price type

An ONIX 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	List 58
Reference name	<DefaultPriceTypeCode>
Short tag	<m185>
Example	*01*

MH.22 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	List 96
Reference name	<DefaultCurrencyCode>
Short tag	<m186>
Example	*USD*

MH.23 Default linear unit DEPRECATED

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	List 94
Reference name	<DefaultLinearUnit>
Short tag	<m187>
Example	*mm*

MH.24 Default unit of weight DEPRECATED

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	List 95
Reference name	<DefaultWeightUnit>
Short tag	<m188>
Example	*oz*

MH.25 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/2.1/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/2.1/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 all ONIX applications, a number of XML attributes may be used where applicable to carry information about the content of an associated element. 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, eight attributes are now defined throughout ONIX, as shown in the tables below. In theory, *any of these attributes can be used in any ONIX element*. In practice, some of them are plainly meaningless except when they are applied to a text element or to a composite that includes text data elements.

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	List 34

Text case	
Function	Enables the case of any text element to be specified.
Form	**`textcase="`**`code`**`"`**
Code list	List 14

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	List 74

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	List 3

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**"**

Attributes are carried within an XML start 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">**.

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 a limited number of 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.

From Release 2.1 Revision 02 onwards EDItEUR is maintaining formal definitions of ONIX both in DTD and XML Schema format. The XML Schema definitions include a complete set of definitions of the code sets used both by data elements and attributes. Users who need a standard mechanism for validating code values are recommended to consider using the XML Schema for this purpose.

5. Start of message "declarations"

At the start of an ONIX Product Information Message, there may be a series of XML "declarations": the *XML declaration*, the *DTD declaration*, and the *namespace declaration*. None of these is mandatory, although it is strongly recommended that a DTD declaration is always included, unless the originating and receiving parties have agreed upon the inclusion of an XML Schema reference instead of a DTD declaration.

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

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

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 for your message that includes characters not listed at the start of section 6.1, you will need to use a modified form of XML declaration as detailed 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 Message 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 all ONIX DTDs 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 the current version of Release 2.1 is:

`http://www.editeur.org/onix/2.1/short/onix-international.dtd`

If the reference names are used, the URI for the current version of Release 2.1 is:

`http://www.editeur.org/onix/2.1/reference/onix-international.dtd`

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 "2.1" being replaced by "2.2" in the above URI.

In the event of intermediate minor revisions being made to the DTD between major releases, these revisions will replace any previous versions of the current release, and the latest revised versions will always be found at the two URIs specified above. However, to ensure that a complete history is always available for any users who need to track it in detail, each successive minor revision will also be held separately on the EDItEUR website at a URI in the following form:

`http://www.editeur.org/onix/2.1/nn/reference/onix-international.dtd`

where "nn" is a 2-digit revision number commencing with "00" for the original issue of this release and "01" for the first revised version. In this way all versions of the current release will remain accessible on the EDItEUR website, at least until the next major release. However, it is strongly recommended that only the latest version of the current release should be used to validate ONIX messages.

Superseded versions should only be used for resolving issues of whether or not an ONIX implementation conforms to the current or to a previous version of the DTD.

Thus, if no minor revisions have been issued for a particular release, there will be only one version of the DTD on the EDItEUR website, at a URI of the form:

`http://www.editeur.org/onix/2.1/reference/onix-international.dtd`

If a first minor revision has been issued as revision 01, the revised DTD will still be at the URI

`http://www.editeur.org/onix/2.1/reference/onix-international.dtd`

but the original Release 2.1 DTD for reference names will be at the URI

`http://www.editeur.org/onix/2.1/00/reference/onix-international.dtd`

and the revised DTD will also be at the URI

`http://www.editeur.org/onix/2.1/01/reference/onix-international.dtd`

The complete standard DTD declaration for the short tag name version of Release 2.1 of the ONIX Product Information Message DTD should contain the following sequence of characters:

```
<!DOCTYPE ONIXmessage SYSTEM
"http://www.editeur.org/onix/2.1/short/onix-international.dtd">
```

The complete standard DTD declaration for the reference name version of Release 2.1 of the DTD should contain the following sequence of characters:

```
<!DOCTYPE ONIXMessage SYSTEM
"http://www.editeur.org/onix/2.1/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 24 files. Only the one referred to above as "*onix-international.dtd* " is referred to directly by the DTD declaration. This file contains a reference to the main tag definition file "*onix-international.elt*", which in turn refers to the XMTHL tag definition file "*onix-xhtml.elt*" and to 18 special character entity definition files. The remaining two special character entity definition files are referred to from within the XHTML tag definition file.

5.3 Namespace declaration

The use of namespaces may be useful in circumstances where ONIX data is being shared with or processed by other XML applications.

The use of a namespace declaration in an ONIX message will render the message invalid according to the DTD, so namespaces should only be used when including ONIX tagged data in other XML applications in accordance with the tagging rules (schemas) for those applications.

If you are using the short tag names, the namespace URI should be:

`http://www.editeur.org/onix/2.1/short`

If you are using the reference names, the namespace URI should be:

`http://www.editeur.org/onix/2.1/reference`

Please note that these URIs do NOT correspond to an actual Web address that is reachable by a browser. They are simply a device for creating an unambiguous reference to the authority for the defined ONIX namespaces.

These namespace URIs are the same as are specified in the corresponding ONIX XML schemas.

5.4 Start and end tags

Following the start of message declarations, the body of the message opens with a start tag:

`<ONIXMessage>` if reference names are used

`<ONIXmessage>` if short tags are used

The message ends with an matching end tag:

`</ONIXMessage>` if reference names are used

`</ONIXmessage>` if short tags are used

5.5 XML Schema references

If, and only if, both originating and receiving parties have agreed to it, the DTD declaration as described in Section 5.2 above may be omitted and an XML Schema reference included instead. It is recommended that, when they leave the originator's system, ONIX messages using the XML Schema should include the appropriate pair of namespace URI and schema location URI, as follows:

ONIX version	*namespace URI* *schema location URI*
Reference	http://www.editeur.org/onix/2.1/reference http://www.editeur.org/onix/2.1/ONIX_BookProduct_Release2.1_reference.xsd
Short	http://www.editeur.org/onix/2.1/short http://www.editeur.org/onix/2.1/ONIX_BookProduct_Release2.1_short.xsd

Further information on the use of XML Schema should be obtained from the website of the World Wide Web Consortium (W3C).

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 as 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*.

The ONIX DTD defines names for a wide range of special characters that can be encoded using XML tags. The special characters are grouped into the following sets:

XHTML 1.0 Special characters (xhtml-special.ent)
XHTML 1.0 Symbols (xhtml-symbol.ent)
ISO Latin characters (iso-lat1.ent and iso-lat2.ent)
ISO Greek symbols (iso-grk3.ent)
ISO Cyrillic alphabetic characters (Iso-cyr1.ent and iso-cyr2.ent)
ISO Numeric and Special Graphic characters (iso-num.ent)
ISO Diacritical Mark characters (iso-dia.ent)
ISO Publishing characters (iso-pub.ent)
ISO Box and Line Drawing characters (iso-box.ent)
ISO General Technical characters (iso-tech.ent)

ISO Additional Mathematical Symbols:
- Ordinary (iso-amso.ent)
- Binary and Large Operators (iso-amsb.ent)
- Relations (iso-amsr-ent)
- Negated relations (iso-amsn.ent)
- Arrow relations (iso-amsa.ent)
- Opening and Closing Delimiters (iso-amsc.ent)

ISO Math Alphabets:
- Fraktur (iso-mfrk.ent)
- Open Face (iso-mopf.ent)
- Script (iso-mscr.ent)

Starting with Release 2.1 the XHTML 1.0 Latin-1 special character set is no longer included in the DTD, as it entirely duplicates special characters in the ISO Latin-1 special character set. The ISO sets included here are the same as the sets implemented in the MathML 2.0 DTD (see http://www.w3.org/ for further information).

These provide a basis for encoding as XML expressions a large number of commonly-used and less 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;
```

In earlier releases of ONIX it was recommended that XML entity references be used in preference to Unicode character references. This recommendation only applies if the ONIX message contains a DTD declaration. Otherwise special characters not available in the encoding character set may only be represented by Unicode character references, ie tags similar to entity references but containing Unicode character numbers instead of entity names.

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
&	`&`
<	`<`

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

```
AT&amp;T
```

7. Using XHTML, HTML, SGML or XML in ONIX text fields

7.1 Introduction

Given the frequent requirement for ONIX messages to convey product information in a form suitable for use in web pages, guidance is provided below on how to incorporate web content in an ONIX product record. However, in order to apply this guidance correctly, a user must already have some knowledge of different forms of web content. Those already familiar with the differences between HTML and XHTML may skip this section.

Web content that is largely text-based is generally styled for presentation in a web page using the HyperText Markup Language (HTML). HTML has been the language of the World Wide Web since its inception and is still the most popular language for constructing web pages. HTML was based upon the Standard Generalized Markup Language (SGML), which has been in use for preparing electronic content in academic and professional publishing since the early 1990s.

XML was developed in the late 1990s as demand grew for ways to use the web for exchanging data and messages that didn't have to be presented as human-readable web pages. XML is a much stricter language than SGML, so it is generally not possible to incorporate HTML-tagged content directly into an XML message. Responding to demand to make it possible to embed HTML in XML, the World Wide Web Consortium has defined an XML-compatible version of HTML, called XHTML. XHTML text fragments *can* be embedded in XML messages, provided this is allowed by the tagging rules of the XML application.

The tagging rules of ONIX specify that XHTML text fragments may be embedded in appropriate ONIX data elements, but within very strict constraints. These constraints are set out in the next section.

HTML text fragments – and indeed any fragment of plain or tagged text, regardless of the tagging language – can be embedded in ONIX data elements, but only by using XML techniques that 'hide' these fragments from any XML-aware software that is processing the ONIX message. Two methods for embedding HTML in an ONIX data element are describe below. These methods are available by default in all XML applications, and cannot be prohibited in ONIX applications, but their use is strongly discouraged.

All ONIX users are encouraged to convert HTML text fragments to be valid XHTML fragments before incorporating them in ONIX messages.

7.2 XHTML (Version 1.0 or later)

The ONIX Product Information Message DTD enables the inclusion of XHTML-tagged text in specific data elements where this has been deemed appropriate. In these cases the data element may contain any well-formed fragment of XHTML-tagged text with the following restrictions:

(a) It must be the case that, if the fragment were to be placed in an otherwise empty <body> element in an XHTML document, the resulting document would be valid;

(b) The fragment may not include any XHTML forms, script or document revision elements;

(c) The fragment may not use “event” attributes and others that may affect browser behaviour.

The intention of these restrictions is to prevent the unwitting or malicious transmittal of viruses in ONIX messages.

The use of XHTML tags in any data element should be signalled by specifying the value of the “textformat” attribute to be "04" (see Section 4) in the start tag of the data element in question, eg:

```
<Text textformat=”04”>... XHTML-tagged text ...            ...</Text>
```

NOTE – The use of ISO special character tags as described in Section 6 is valid in data elements that contain XHTML tags, provided the ONIX message contains a DTD declaration. However, if an XHTML-tagged fragment containing such special character tags is copied from an ONIX message into an XHTML document, the special character tags should be replaced with the corresponding Unicode character references as defined in the relevant modules of the DTD – otherwise the resulting XHTML document will generally be invalid.

The elements in the ONIX Product Information Message for which XHTML tagging is enabled are:

PR.7	<WebsiteDescription> (and in all occurrences of this element in other contexts)
PR.8	<BiographicalNote>
PR.15	<Annotation> <MainDescription <Text> <ReviewQuote>
PR.16	<TextWithDownload> <DownloadCaption> <DownloadCredit> <DownloadCopyrightNotice> <DownloadTerms> <ProductWebsiteDescription>
PR.17	<PrizeJury>

7.3 HTML (Version 4.0 or earlier)

The inclusion of text tagged in accordance with HTML version 4.0 or earlier in an ONIX data element is possible using one of the two methods described below for SGML-tagged text, but this is now deprecated. In the event that HTML is included in either of these ways, the "textformat" attribute on the start tag for the element should be specified with the value "02" (HTML, other than XHTML).

7.4 Other SGML or XML

There are two methods for including SGML- or XML-tagged text in an ONIX data element.

Method 1. The "textformat" attribute on the start tag for the element should be specified with the value "01" (SGML) or "03" (XML). The "**<**" character at the start of every SGML or XML element start and end tag is replaced by its entity reference as described in the previous section. For example:

```
<Text textformat="03">&lt;para>XML-tagged review quotation...
&lt;emph>emphasized text&lt;/emph> ...&lt;/para></Text>
```

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

Method 1 is the *recommended* method for including SGML- or XML-tagged data.

NOTE – The technique of replacing the "**<**" symbol by "**<**;" 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 "**é**" replaced by "**é**"), 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.

Method 2. The "textformat" attribute on the start tag for the element should be specified with the value "01" (SGML) or "03" (XML). 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[<para>XML-tagged review quotation...
<emph>text in italics</emph> ...</para>]]></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 SGML- or XML-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 2 is *not* recommended.