NewsML™ 1.2

Guidelines V 1.00

Guidelines Version 1.0 / 2004-08-18

NewsML™ 1.2 – Guidelines V 1.00

Document file name: NewsML_1.2-doc-Guidelines_1.00

Document URN: urn:iptc:std:newsml:1.2:doc:guidelines:1.00

Revision history:

Revision	Issue Date	Published by	Remark
1.00	2004-08-18	Laurent Le Meur & Michael Steidl, MD	First version of fully revised Guidelines documentation.

If you find any problems in this documentation, please report them back to the IPTC: office@iptc.org

Find updated information on NewsML at: www.newsml.org

IPTC company contact:
IPTC
Royal Albert House,
Sheet Street, Windsor,
Berkshire, SL4 1BE (UK)
Phone: +44 (0)1753 705051
Fax: +44 (0)1753 831541
Email: office @iptc.org
Web: www.iptc.org

The IPTC is registered in England at Windsor House, Victoria Street, Windsor, SL4 1EN
as Comité International des Télécommunications de Presse
Registration No. 1010968 Limited by Guarantee Not Registered for VAT

NewsML 1.2 - Guidelines
Table of contents

scratch. Discover how controlled vocabularies (e.g. TopicSets) are referenced from NewsML instances, and how local sets of Topics may also be created inside NewsML instances for specific purposes.

9. Extension mechanisms

See how NewsML may be extended, via the extension of existing TopicSets or the creation of new TopicSets, the creation of new types of Property elements in existing metadata sets or the creation of new types of Metadata sets.

10. Appendix

Get extra guidelines about XML encoding, the format of dates, the NewsML namespace URI, the proper validation of NewsML instances and the versioning of the NewsML standard itself.

NewsML 1.2 - Guidelines
Chapter 1: Introduction

1 Introduction

1.1 NewsML

The purpose of NewsML is to make possible the exchange of news – whether text, photos or other media – accurately and quickly by bundling the content in a way that allows highly automated processing. The core technology is XML.

News exchange is the process of moving around not only the core news content, but also data that describe the content in an abstract way (i.e. metadata), information about how to handle news in an appropriate way (i.e. news management data), and finally information about the news transportation or routing process (i.e. exchange data).

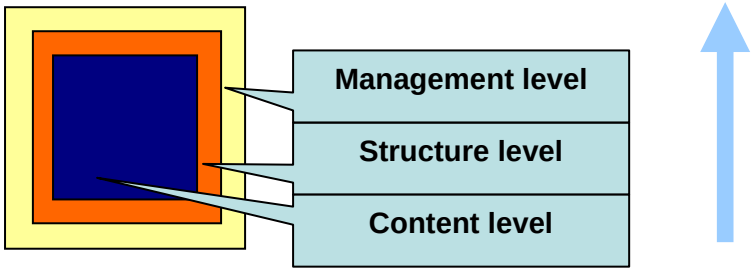
Having a strong background in developing and maintaining news exchange formats, the IPTC created NewsML as the most comprehensive and versatile way to move all these types of data between media-handling systems to make news exchange efficient and reliable.

1.2 Structure of NewsML

NewsML has a four-level structure. Each level is represented by a ***news object***. The definition of a news object, found in the NewsML specifications, is one of the main constituents of NewsML documents. The different kinds of news objects are NewsEnvelope, NewsItem, NewsComponent and ContentItem.

These levels provide complementary functions and are labelled from outermost to innermost:

1. The **Exchange** level - the **NewsML** document level element, along with the **NewsEnvelope** element;
2. The **Management** level – primarily represented by the **NewsItem** element;
3. The **Structure** level – primarily represented by the **NewsComponent** element;
4. The **Content** level – primarily represented by the **ContentItem** element.



The following chapters of the guidelines describe this model from the innermost tier outwards to provide an incremental introduction to NewsML functionality.

1.3 Goal of this document

A. Create a fundamental understanding of the structures and functions within a NewsML instance and add examples so that an implementer of a NewsML system gets a guideline for making decisions and choosing the right tools and technology.

B. Give recommendations and best practices; offer a path to consistent implementations of complex information structures.

1.4 Audience

System architects, project managers, software developers with a knowledge of XML technology and a basic understanding of the news industry.

NewsML users can be providers of news, consumers of news or news system providers.

1.5 Related documentation

These guidelines are part of a broader set of comprehensive explanatory material that describes NewsML and other IPTC standards. In particular, they complement the "**NewsML Functional Specification**" which together with the NewsML 1.2 DTD forms the definitive reference for NewsML. This document is intended to amplify upon the prescriptive material and contains additional descriptive, tutorial and best practice information.

The guidelines refer to the NewsML Functional Specifications for unusual features, and complementary information is found in other parts of the NewsML documentation, named "**Common Implementations**" and "**The Expert Zone**".

1.6 Acknowledgements

This documentation is the result of a team effort by members of the International Press Telecommunications Council, with input and assistance from others.

Particular contributions are as follows:

The documentation was edited by Laurent Le Meur (Agence France Presse). The work was overseen by the IPTC Managing Director Michael Steidl, and incorporates work by several IPTC members, notably Jo Rabin (linguafranca.org), Stuart Myles (wsj.com), Johan Lindgren (Tidningarnas Telegrambyrå), Harald Löffler (IFRA), Takahiro Fujiwara (EAST Co., Ltd.) and the other members of the Japanese NSK NewsML team.

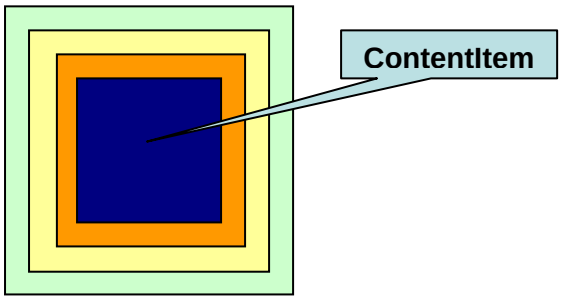
Copy editing was done by Walter Baranger (New York Times).

NewsML 1.2 Guidelines
Chapter 2: The Content Level

2 The content level – ContentItem

2.1 Context of the ContentItem

The core level of the NewsML Model, the content level, provides a uniform interface to content irrespective of the media type of that content.



A ContentItem is the unit of data content managed in a news environment; it can represent a block of enriched text, a JPEG photo, an EPSF graphic, a Flash animation, a video clip or any other type of media asset.

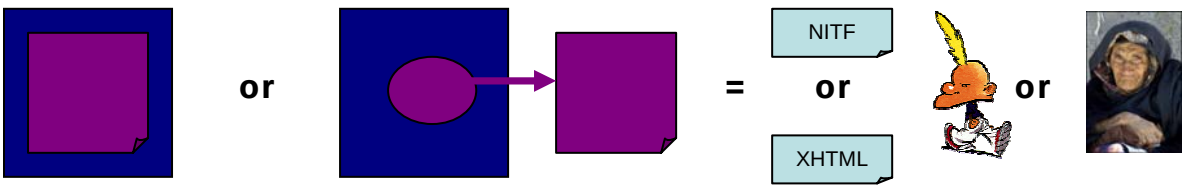
2.2 Structure of a ContentItem

The ContentItem element is a wrapper that makes available to the NewsML processor both the content itself and metadata describing the technical properties or physical characteristics of the content.

ContentItems are never found as stand-alone pieces of information in a news environment; they are instead included in NewsComponents, where the role they play in contributing to the NewsItem as a whole is specified.

ContentItems can make their content available in two distinct ways:

1. by including it inline within the ContentItem element
2. by providing a reference to the content so that the NewsML application can retrieve it from someplace else, such as a remote server.

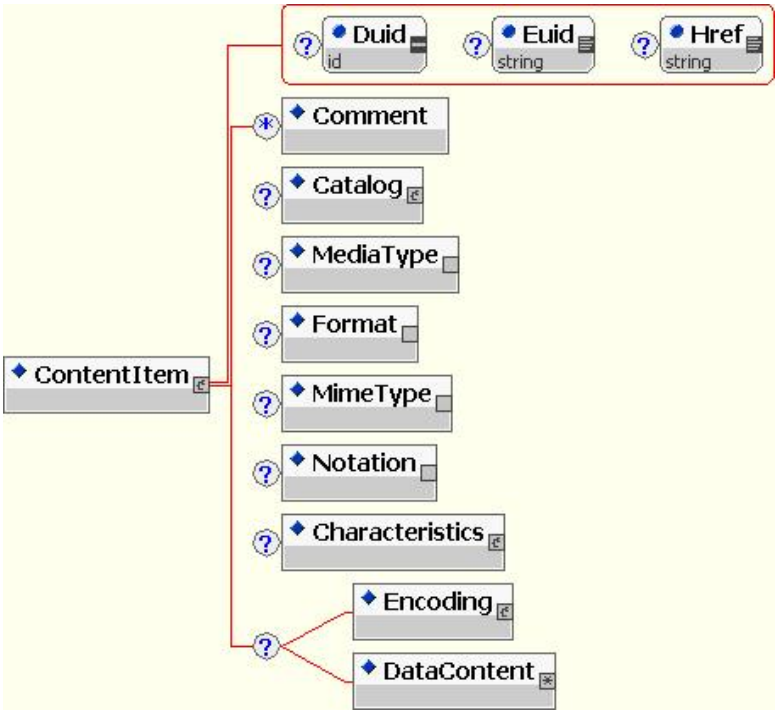


The metadata surrounding the ContentItem is quite often information that is available within the content itself. The purpose providing metadata at the ContentItem level is twofold:

a. to make it available to the NewsML application by use of an XML parser – i.e. without having to launch an application to interpret the content which is often represented in a non-XML format, and to allow easier comparison between different content formats.

b. so that NewsML applications do not have to retrieve referenced content to determine its characteristics. This saves time and bandwidth.

The different types of content metadata provided by NewsML are a media type, a format, mime type, notation and a set of characteristics.



2.3 Content inclusion

The data of a ContentItem can be referenced via its Href attribute, or included explicitly in its **DataContent** sub-element. DataContent is of type ANY; that means that any XML compatible data can be inserted in this element; this includes encoded binary content.

When textual content is included in the DataContent element, its character set must be the same as the character set of the including NewsML document; UTF-8 is recommended.

2.3.1 Reference to content

The **Href** attribute value is a URL (Uniform Resource Locator) that indicates *where* the remote content can be accessed.

This is the recommended method when the content is some binary data.

Example 2.3.1:

```
<ContentItem Href="http://www.mycompany.com/images/BER90-082802a.jpg">
    <MediaType FormalName="Photo"/>
    <Format FormalName="JPEG Baseline"/>
    <Characteristics>
        <Property FormalName="Width" Value="2048"/>
        <Property FormalName="Height" Value="1543"/>
    </Characteristics>
</ContentItem>
```

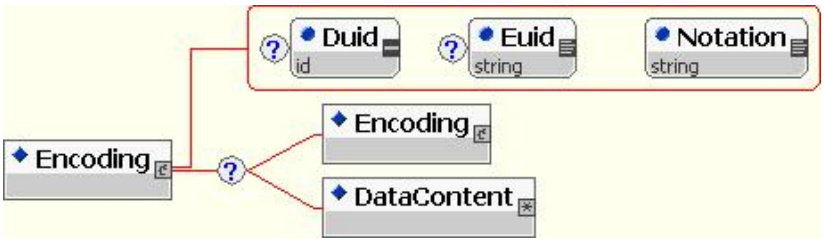
In this sample, the data is available via an http URL, but a relative URL is also possible. The simplest link is of the form <ContentItem Href="./BER90-082802a.jpg"> when the NewsML instance and the linked resource are found in the same directory.

Note about URLs:
Any kind of URL (e.g. ftp: or https:) is allowed as Href value. A URL *isn't* a global i.d.; a resource can be freely duplicated, so identical resources can be referenced by different URL's. Even if not duplicated, a single resource can be pointed at by different URL's, using alternative syntaxes (e.g. IP numbers vs host names). Users are also reminded that the referenced resource can be shared between several news items without any warning in NewsML.

Note about the use of Encoding:
An empty Encoding element (or a cascade of Encoding elements – see section 2.3.2 – ending with an empty one) may be inserted in order to represent the encoding of the remote file. In this case the Format, MimeType or Notation elements are relative to the format of the remote data before encoding (to avoid discrepancy with the inline inclusion of binary content; see below).

2.3.2 Inline inclusion of binary content

Binary content can be in-lined in a DataContent element. An **Encoding** element wraps DataContent in such a case; the **Notation** attribute of the Encoding element indicates the encoding algorithm (e.g. base64, binhex).



The IPTC maintains a basic set of encoding values and descriptions on the IPTC Web site.

Note about associated properties:
The Format, MimeType or Notation properties are relative to the data before encoding.

Note about the semantics of Notation:
The Notation attribute of Encoding and the Notation element in ContentItem (see below) do not have the same semantics.

Example 2.3.2, a JPEG image encoded in Base64:

```
<ContentItem>
    <MediaType FormalName="Photo"/>
    <Format FormalName="JPEG Baseline"/>
    <Encoding Notation="base64">
        <DataContent>SDF35GZsDFGZER5R6RTGSE5 ... SRTYERT</DataContent>
    </Encoding>
</ContentItem>
```

Where multiple encoding has been applied -- such as a Zip file that may have been subsequently Base64 encoded -- then multiple Encoding elements are used as follows:

Example 2.3.2a, a Freehand graphic with Zip+Base64 encoding:

```
<ContentItem>
    <MediaType FormalName="Graphic"/>
    <Format FormalName="Freehand"/>
    <Encoding Notation="base64">
        <Encoding Notation="zip">
            <DataContent>SDF35GZsDFGZER5R6RTGSE5 … SRTYERT</DataContent>
        </Encoding>
    </Encoding>
</ContentItem>
```

The last applied encoding is designated in the Encoding element that wraps any other previous encoding that may have taken place. Freehand encoded as Zip[Freehand] encoded as base64[Zip[Freehand]] in the previous example.

2.3.3 Inline inclusion of textual content

Several options allow for the inline inclusion of textual data in a DataContent element.

Preferred options are listed first.

2.3.3.1 Use of a namespace

A specific namespace can be declared within the DataContent element.

This is useful when the document is to be validated via an XML schema. But no XML validation can be applied via a DTD if the "xmlns" attribute is not defined in the dtd associated with the content payload.

Example 2.3.3.1:

```
<ContentItem>
    <MediaType FormalName="Text"/>
    <Format FormalName="NITF"/>
    <DataContent>
        <nitf xmlns="urn:newsml:iptc.org:20011012:NITF">
            <body>
                <body.content>
                    <p>Today, <person>Clinton</person> visited…</p>
                    <p><person>Al Gore</person> also attended the…</p>
                </body.content>
            </body>
        </nitf>
    </DataContent>
</ContentItem>
```

Recommendation:
The element inserted in the DataContent element should be the root element of the included XML structure (e.g. <nitf>, <xhtml>, etc.).

Going deeper:
More information about validation using XML schemas is given in **Chapter 10.4**.

2.3.3.2 Direct inclusion without namespace declaration

The inclusion of inline XML data without namespace declaration is possible.

In this case an XML validation can be applied via a DTD. To do so, the included must be declared as an entity reference or as a local DTD subset in the DOCTYPE declaration structure at the top of the document instance.

Example 2.3.3.2:

```
<!DOCTYPE NewsML SYSTEM "http://www.company.com/dtd/NewsML_1.2.dtd" [
<!ENTITY % NITF SYSTEM "http://www.company.com/dtd /nitf-3-1.dtd">
%nitf;
]>
<NewsML>
    <!-- --------- -->
    <ContentItem>
        <MediaType FormalName="Text"/>
        <Format FormalName="NITF"/>
        <DataContent>
            <nitf>
                <body>
                    <body.content>
                        <p>Today, <person>Clinton</person> visited…</p>
                        <p><person>Al Gore</person> also attended the…</p>
                    </body.content>
                </body>
            </nitf>
        </DataContent>
    </ContentItem>
    <!-- --------- -->
</NewsML>
```

2.3.3.3 Use of a CDATA section

It is possible to keep the text in original and insert it within a CDATA section so that it is completely opaque to the NewsML processor.

This is especially useful for HTML (non XML) inclusion, but it is *not* recommended for XML inclusion.

Example 2.3.3.3:

```
    <ContentItem>
        <MediaType FormalName="Text"/>
        <Format FormalName="HTML"/>
        <DataContent>
            <![CDATA[
            <html>. . .</html>
            ]]>
        </DataContent>
    </ContentItem>
```

Inclusion of content using a CDATA section works only if the content does not itself contain a CDATA section (or contain the character sequence "]]>") for any other reason. Unless you are sure this is the case it is best to avoid it. For the sake of consistency between different vendor implementations of NewsML, it is strongly recommended that CDATA sections are not used to embed XML documents – the receiver's processing of a CDATA section that contains an XML document may differ from one that contains text.

2.3.3.4 Plain text inclusion

Ensuring there are no "mark-up" characters present, insertion of plain text is the last supported solution.

Example 2.3.3.4:

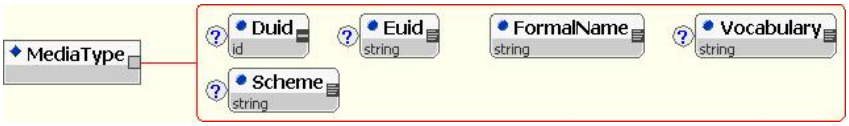
```
    <ContentItem>
        <MediaType FormalName="Text"/>
        <MimeType FormalName="text/plain"/>
        <DataContent>The quick brown fox jumped over the lazy dog. The lazy dog &amp;amp; his aunt showed quite some surprise! “That’s more than (&amp;gt;) I was bargaining for,” she said.</DataContent>
    </ContentItem>
```

2.4 Properties of a ContentItem

2.4.1 Media type

ContentItems fall into several categories. The media type is identified in the **MediaType** sub-element through the use of the FormalName attribute and its associated Vocabulary and Scheme attributes. In this chapter, the controlled values representation is restricted to the FormalName attribute.

> **Going deeper:**
> More information about NewsML controlled vocabularies, vocabulary attributes (Scheme and Vocabulary) and topic sets is given in **Chapter 8**, "**NewsML Controlled Vocabularies**".



Current values defined by the IPTC are:

- **Text** : represents a text body, e.g. a wire story, an article content, or a report. Such a string can be a plain text, or an XML structure with internal markup.
- **Graphic** represents a still or animated graphic, in a bitmap or vector form. A dynamic graphic can be a 2D or 3D work, and can be interactive, or a pure animation
- **Photo** represents a digital picture, a snapshot of a real world situation.
- **Audio** represents a digital audio sequence.
- **Video** represents a digital video sequence; the sequence may include a soundtrack.
- **ComplexData** represents a software application or other composite data File.

Those media types are defined in a "topic set" maintained by the IPTC on its Web site.

> **Recommendation**:
> The inclusion of a media type is strongly recommended for any ContentItem represented in NewsML.

2.4.2 Format, MimeType and Notation

ContentItems can also have a **Format**, a **MimeType** and a **Notation**. Those elements have the same structure as MediaType, and the values of those properties are identified through the use of a FormalName attribute and associated vocabulary attributes.

The purpose of those elements is to allow the receiver to assign an appropriate application to process it, or add an appropriate file extension, so that the content is associated with the correct application on their system. As such, those three elements can be considered as alternatives ways of conveying the same information. The information is relative to the data content before any transformations described by the Encoding element have been applied (see "**Reference to content**" and "**Inline inclusion of binary data**").

The **Format** vocabulary is maintained by the IPTC on its Web site.

Examples of Format are “NITF”, “XHTML”, “JPEG Baseline”, “Waveform Audio”, “FLA”, “MPEG”.

The **MimeType** vocabulary is maintained by the Internet Assigned Numbers Authority of California, U.S.A.. The IPTC maintains a set of the most useful mime type values on its Web site.

Examples of MimeType are “**text/vnd.IPTC.NITF+xml**”, “**image/jpeg**”, “**audio/x-wav**”, “**video/mpeg**”.

The **Notation** property is a third alternative way of giving information about the content structure. The IPTC defined three alternative schemes for this vocabulary: an IPTCNotation scheme, a Formal Public Identifier scheme, and a NewsML URN scheme. The IPTC maintains a set of the most useful notations values on its Web site.

Examples of Notation are “NITF” (IptcNotation scheme), “-//IPTC-NAA//DTD NITF-XML 1.0//EN” (Formal Public Identifier scheme), “urn:newsml:iptc.org:20001006:NITF” (NewsML URN scheme).

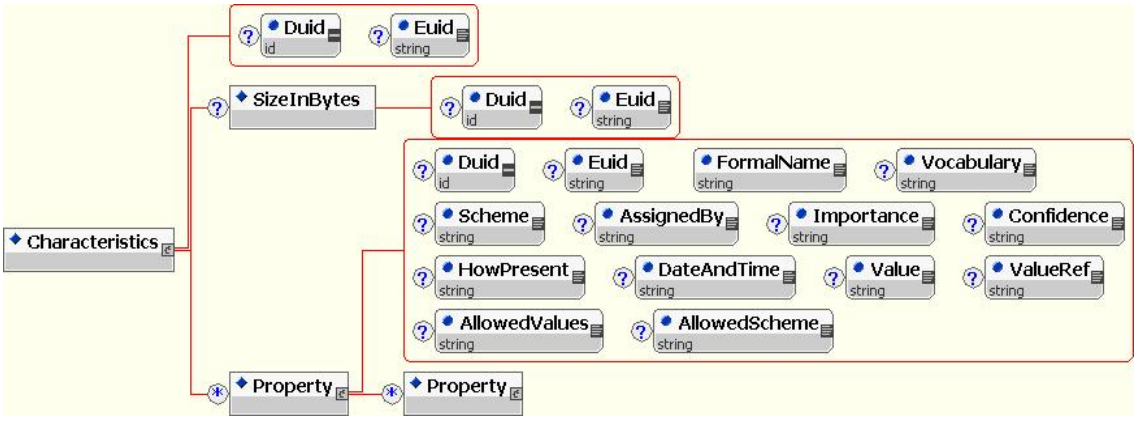
Recommendation:
The inclusion of a Format or a MimeType is strongly recommended for any ContentItem represented in NewsML. If both are present, they should be equivalent in their meaning; if not, the indication given by Format takes precedence.

Note about MimeType vs Format:
- The MimeType property is somewhat redundant with Format, and registering a new mime type in the IANA registration mechanism is sometimes difficult, so some useful formats lack a mime type value.
- Note also that the Format and MimeType IPTC TopicSets are not perfectly in sync.

Note about Notation:
Notation comes from the SGML world, and few implementers use it. The Notation information should be used when no Format or MimeType indication is available.

2.4.3 Characteristics of content

A ContentItem supports different optional **Characteristics**.



The IPTC has left it to the publishers to decide the best way of employing them for precisely defining the physical characteristics of content. Ideally, say just enough about a piece of content to tell a subscriber how to process it for appropriate rendering.

A **SizeInBytes** is the only physical metadata currently defined by the IPTC, as this property is common to all media types. Other characteristics are specific to certain media types; they can be freely added by providers using a generic **Property** element.

The information is relative to the data content before any inline encoding was applied (see "**Reference to content**" and "**Inline inclusion of binary data**").

> **Going deeper:**
> The generic **Property** element is key to the NewsML extension mechanism; it is illustrated below and fully described in **Chapter 9**, "**Extension Mechanisms**".

2.4.3.1 Format version

The DataContent in a ContentItem may be in a number of different formats. These formats themselves are modified over time and often appear in updated versions. In order to cater for this need to reflect version changes in existing formats it is recommended that the following method of expression is used:

Example 2.4.3.1:

```
<ContentItem>
    <MediaType FormalName="Text"/>
    <Format FormalName="NITF"/>
    <Characteristics>
        <Property Scheme="CharacteristicsProperty" FormalName="FormatVersion" Value="3.1"/>
    </Characteristics>
    <DataContent >
        <nitf>……</nitf>
     </DataContent>
</ContentItem>
```

The version is appended through the Characteristics element with a child Property assigned a FormalName ="FormatVersion" attribute. This controlled value comes from a Topic Set with a Scheme named "CharacteristicsProperty". The Value attribute data (not drawn from a Topic Set) is the version number.

2.4.3.2 Characteristics of different media

The IPTC maintains a set of useful media characteristics values and descriptions.

The following guidelines list the characteristics applicable to the media used in news reporting, and give some examples of use of those properties.

2.4.3.2.1 Text

Applicable characteristics are: **Words, Alphabet, Font**.

2.4.3.2.2 Photo

Applicable characteristics are: **Quality level, Width, Height, ColorSpace, PixelDepth, Rotation, ICCProfile**.

Example 2.4.3.2.2:

```
<ContentItem Href="http://www.company.com/photo.jpg">
    <MediaType FormalName="Photo" />
    <Format FormalName="JPEG Baseline" />
    <Characteristics>
        <SizeInBytes>401230</SizeInBytes>
        <Property FormalName="Width" Value="3072"/>
        <Property FormalName="Height"  Value="2536" />
        <Property FormalName="ColorSpace"  Value="RGB"/>
        <Property FormalName="ICCProfile"  Value="Nikon D1"/>
    </Characteristics>
</ContentItem>
```

2.4.3.2.3 Graphic

Applicable characteristics are: **AnimationType, Resolution, HeighWidthRatio**.

In the case of a bitmap graphics, some characteristics are shared with photos like **Height** and **Width**. In the case of a non interactive animation (e.g. Macromedia Flash graphics), a **Duration** can be inserted.

2.4.3.2.4 Audio

Applicable characteristics are: **AudioCoder, AudioCoderVersion, TotalDuration, AverageBitRate, SampleSize, SampleRate, AudioChannels**.

Example 2.4.3.2.4:

```
<ContentItem Href="http://www.company.com/audio-report.mp3">
    <MediaType FormalName="Audio" />
    <Format FormalName="MP3" />
    <Characteristics>
        <SizeInBytes>123456</SizeInBytes>
        <Property FormalName="FileExtension" Value=".mp3"/>
        <Property FormalName="AudioCoder"  Value="MP3" />
        <Property FormalName="AudioCoderVersion" Value="1.0"/>
        <Property FormalName="TotalDuration"  Value="30"/>
        <Property FormalName="AverageBiteRate"  Value="128"/>
        <Property FormalName="SampleSize"  Value="16"/>
        <Property FormalName="SampleRate"  Value="44.100"/>
        <Property FormalName="AudioChannels"  Value="2"/>
    </Characteristics>
</ContentItem>
```

2.4.3.2.5 Video

Applicable characteristics are: **VideoCoder, VideoCoderVersion, Vbr, Width, Height, TotalDuration, FramesTotal, FrameRate, KeyFrames, PixelDepth, AverageBitRate, Sampling, Redirector**.

2.4.3.2.6 Controlled values of Characteristics

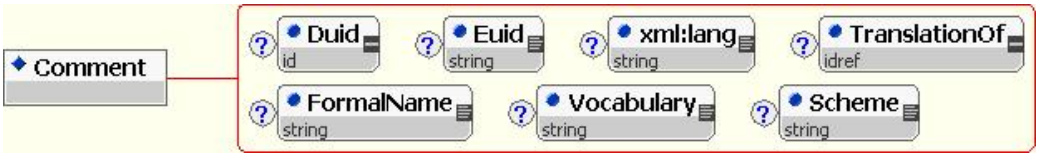
In order to have standard controlled vocabularies for physical characteristics the IPTC has produced additional Topic Sets for the following Characteristics Property values:

- **AudioCoder** ,
- **VideoCoder**,
- **ColorSpace**.

The corresponding topic sets are maintained on the IPTC Web site.

2.5 The Comment element

As many other NewsML elements, ContentItem may have one or more Comment sub-elements, which provide informal additional information in natural language for an editorial audience.



The **xml:lang** attribute identifies the language of the contents of an XML element. It is defined in the XML specification and its value must be as defined in the IETF RFC 3066.

The **TranslationOf** attribute is a pointer to another Comment element, of which this Comment is a direct translation. The pointed element must be locally identified by a Duid attribute, and the pointer takes the form of an 'idref'.

Added in NewsML v1.1, the **FormalName** and associated vocabulary attributes optionally add a type to the comment. The choice of a controlled vocabulary is up to the news provider.

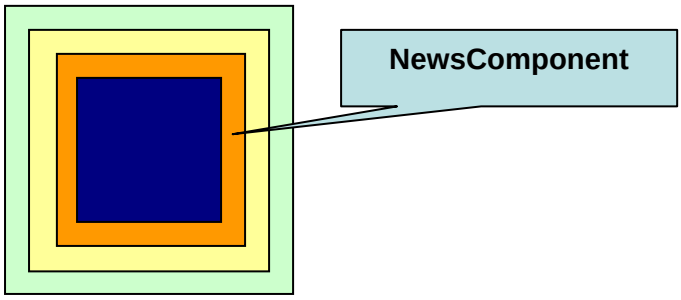
Creator: L. Le Meur
Main contributors: J.Rabin, J.Lindgren, M.Steidl, japanese NSK team

NewsML 1.2 - Guidelines
Chapter 3: The Structure Level

3 The structure level - NewsComponent

3.1 Context of the NewsComponent

Above the content level is the structure level, represented by the NewsComponent.



The NewsComponent serves several purposes in relation to the NewsItem as a whole.

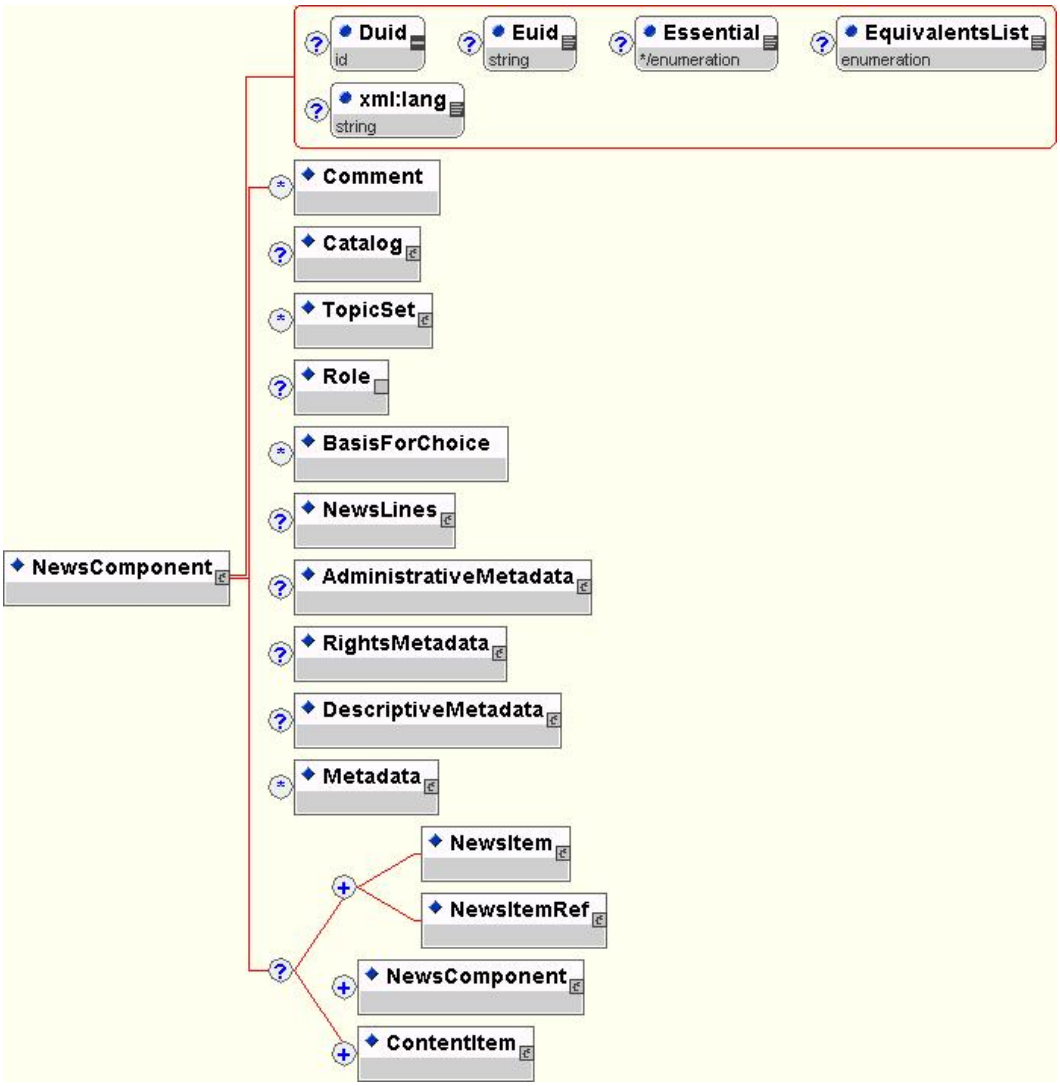
1. It acts as a ***container*** for news objects (i.e. ContentItems, NewsComponents or NewsItems); in the following discussion, those objects are called ***constituents*** of the NewsComponent.
2. It allows the attachment of ***metadata*** that is relevant to, and shared by its constituents.
3. It allows the attachment of ***NewsLines*** which contain human readable information about its constituents.

In this chapter, we describe the NewsComponent as a flexible container for news objects, i.e. ContentItems, NewsComponents, NewsItems or NewsItemRefs.

A detailed description of the NewsComponent used as an element holding metadata about content is given in **Chapter 4**, **"Metadata about content – NewsComponent**".

3.2 Structure of a NewsComponent

A NewsComponent adds structure to composite documents; it groups related news objects in consistent collections, which in some sense belong together. For example a NewsComponent can represent a picture with its caption, or several text parts in different languages.



NewsComponents are not something that can be created (like some content, or a NewsItem); rather, they are like parentheses in a mathematic equation: they glue together objects that share something (they share metadata, or are grouped as a logical collection).

> **Note about NewsItem and NewsEnvelope:**
> NewsItem and NewsEnvelope objects are covered in chapter 5 ("**The management layer – NewsItem**") and chapter 7 "**The exchange layer – News Envelope**").

NewsComponents can represent collections of two kinds: its constituents can be alternatives to each other (i.e. an **equivalents list**) or can be complementary to each other (i.e. a **complementary list**). NewsComponents may also state what role their constituents play in the NewsItem in which they appear. These aspects are discussed in more detail below under chapter 3.32 "Role of objects" and "Equivalence of objects".

A NewsComponent has an implied ordering of its complementary content, which is the order of appearance in the collection – i.e. document order. Explicit ordering may be expressed by a provider using a proprietary mechanism.

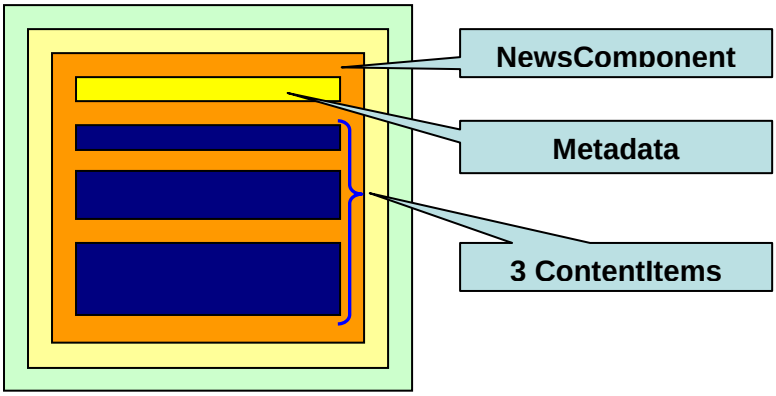
The recursive aspect of those collections of news objects – NewsComponents can contain NewsComponents - allows for the representation of information trees of arbitrary complexity. Database designers should be aware of these implications.

Note about the container model:
During IPTC discussions, some participants thought of a NewsComponent as a replicate of a **MIME envelope**, with a "multipart/related" or "multipart/alternative" model. Other participants compared it to an **RDF container** that could act as a 'Bag' (unordered list), a 'Sequence' (ordered list), or an 'Alternative' list (a list of alternative resources). This idea evolved; the idea of complementary list vs. equivalents list was kept, but there was no decision about the explicit representation of ordered and non-ordered lists.

3.2.1 Objects found in a NewsComponent

A NewsComponent can include NewsItems, NewsItemRefs, NewsComponents or ContentItems. NewsItems and NewsItemRefs can be combined with each other but no other combinations are allowed.

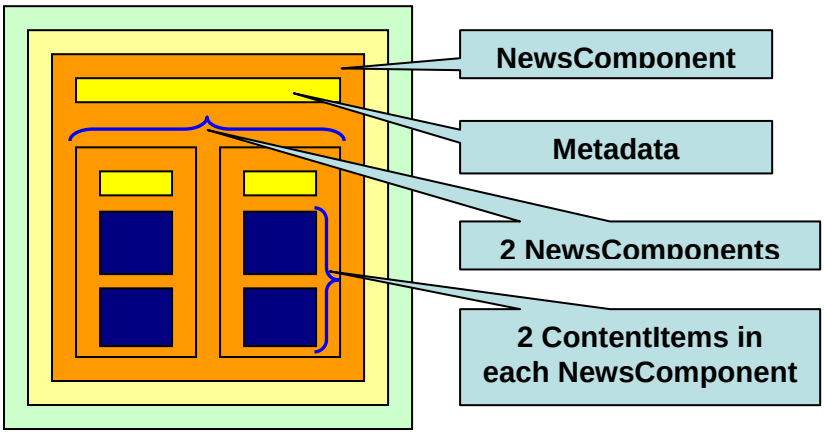
Example 3.2.1: a NewsComponent holds three ContentItems. These can be three pictures in three different definitions (thumbnail, preview and full-resolution images).



Snippet 3.2.1:

```
<NewsComponent>
    <AdministrativeMetadata>
        <Creator>
            <Party FormalName="john.doe@mycompany.com"
Vocabulary="urn:newsml:mycompany.com:20031010:people"/>
            <Contribution FormalName="Photographer"/>
        </Creator>
    </AdministrativeMetadata>
    <DescriptiveMetadata>
        <SubjectCode><Subject FormalName="1506206"/></SubjectCode>
    </DescriptiveMetadata>
    <ContentItem Href="./photo1-thumbnail.jpg">
        <MediaType FormalName="Photo"/>
        <Format FormalName="JPEG Baseline"/>
        <Characteristics>
            <Property FormalName="Width" Value="128"/>
            <Property FormalName="Height" Value="98"/>
        </Characteristics>
    </ContentItem>
    <ContentItem Href="./photo1-preview.jpg">
        <MediaType FormalName="Photo"/>
        <Format FormalName="JPEG Baseline"/>
        <Characteristics>
            <Property FormalName="Width" Value="512"/>
            <Property FormalName="Height" Value="430"/>
        </Characteristics>
    </ContentItem>
    <ContentItem Href="./photo1-full.jpg">
        <MediaType FormalName="Photo"/>
        <Format FormalName="JPEG Baseline"/>
        <Characteristics>
            <Property FormalName="Width" Value="2048"/>
            <Property FormalName="Height" Value="1543"/>
        </Characteristics>
    </ContentItem>
</NewsComponent>
```

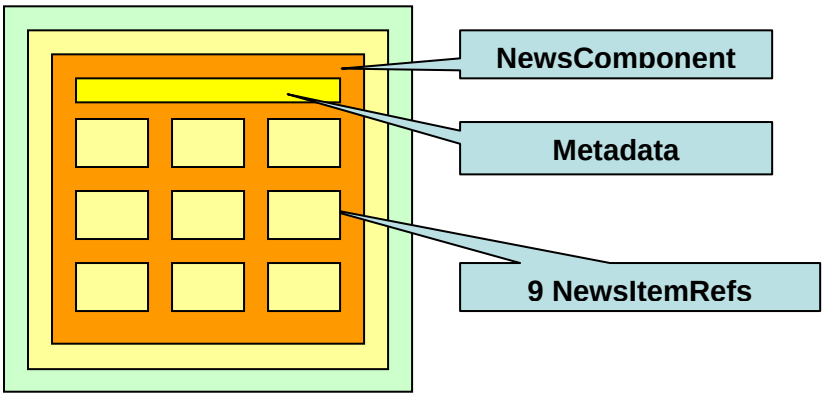
Example 3.2.1a: a NewsComponent holds two NewsComponents; each included NewsComponent holds two ContentItems. This can be a short and a long version of a story in Italian as a NewsComponent holding two equivalents ContentItems, along with a thumbnail and preview size picture as another NewsComponent also holding two equivalents ContentItems.



Snippet 3.2.1a:

```
<NewsComponent>
    <AdministrativeMetadata>
        <Provider>
            <Party FormalName="MyCompany"/>
        </Provider>
    </AdministrativeMetadata>
    <NewsComponent EquivalentsList="yes">
        <AdministrativeMetadata> ...
            <Creator>
                <Party FormalName="Enzo Massimo"/>
            </Creator>
        </AdministrativeMetadata>
        <DescriptiveMetadata>
            <Language FormalName="it"/>
        </DescriptiveMetadata>
        <ContentItem>
            <MediaType FormalName="Text"/>
            <Characteristics>
                <SizeInBytes>266</SizeInBytes>
            </Characteristics>
            <DataContent> ... </DataContent>
        </ContentItem>
        <ContentItem>
            <MediaType FormalName="Text"/>
            <Characteristics>
                <SizeInBytes>1356</SizeInBytes>
            </Characteristics>
            <DataContent> ... </DataContent>
        </ContentItem>
    </NewsComponent>
    <NewsComponent EquivalentsList="yes">
        <Creator>
            <Party FormalName="John Doe" />
        </Creator>
        <ContentItem Href="./photo1-thumbnail.jpg">
            <MediaType FormalName="Photo"/>
            <Format FormalName="JPEG Baseline"/>
            <Characteristics>
                <Property FormalName="Width" Value="128"/>
                <Property FormalName="Height" Value="98"/>
            </Characteristics>
        </ContentItem>
        <ContentItem Href="./photo1-preview.jpg">
            <MediaType FormalName="Photo"/>
            <Format FormalName="JPEG Baseline"/>
            <Characteristics>
                <Property FormalName="Width" Value="512"/>
                <Property FormalName="Height" Value="430"/>
            </Characteristics>
        </ContentItem>
    </NewsComponent>
</NewsComponent>
```

Example 3.2.1b: a NewsComponent holds 9 NewsItemRefs. This can be a list of syndicated NewsItems referred to by their globally unique identifier.



Snippet 3.2.1b:

```
<NewsComponent>
    <AdministrativeMetadata>
        <Provider>
            <Party FormalName="MyCompany"/>
        </Provider>
    </AdministrativeMetadata>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:25ab03:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:26cb02:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:03bg23:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:06kl43:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:58fn34:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:38ke56:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:27hn89:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:46ab34:1"/>
    <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:25av03:1"/>
</NewsComponent>
```

Counter example: it is not possible to include in the same NewsComponent a ContentItem which represents an abstract in a text block (without any administrative or descriptive metadata) and a NewsItemRef which supports the reference of a fully identified article. Both have to be wrapped in NewsComponents, so they can coexist in the same container (sample shown below in chapter 3.3.2 “Equivalence of objects”).

3.2.2 Local identification of a NewsComponent

A local identifier – the **Duid** attribute - is usually associated with a NewsComponent so that it can be pointed to from another part of the document.

In particular, adding a local identifier to a NewsComponent allows for its update in a later revision (see **Chapter 6** "**news management – management strategies**").

Another example of the use of a local identifier is the association of an illustration placeholder in a text (e.g. the NITF "media-reference" element) with the proper illustration represented as a NewsML NewsComponent; the following example shows the use of the media-reference@source attribute (value #photo0) to reference the proper NewsComponent via its Duid attribute.

Example:

```
<NewsComponent>
    <NewsComponent>
        <ContentItem>
            <MediaType FormalName="Text"/>
            <Format FormalName="NITF"/>
            <DataContent>
                <nitf><body><body.content>
                    <media media-type="image">
                        <media-reference source="#photo0"/>
                    </media>
                    <p>...</p>
                </body.content></body></nitf>
            </DataContent>
        </ContentItem>
    </NewsComponent>
    <NewsComponent Duid="photo0">
        <ContentItem Href="./photo0.jpg"/>
    </NewsComponent>
</NewsComponent>
```

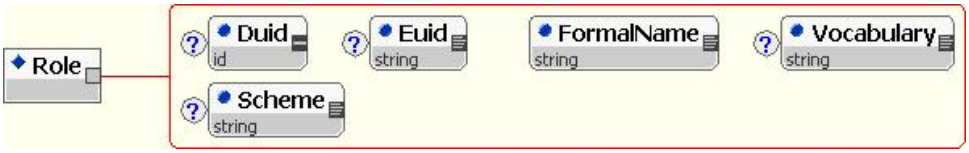
As NewsComponents don't have a globally unique identifier, they are not considered web *resources*, they're not *reusable* in a NewsML environment: there's no way of knowing that two NewsComponents are identical, and no direct way to refer to an external NewsComponent; global identifiers are defined at the NewsItem level (see **Chapter 5** "**news management – NewsItem**").

3.3 Properties of the container

3.3.1 Roles of objects

Each NewsComponent in a news document can contain a **Role** element. A Role is the distinguishing characteristic of a NewsComponent, or its relationship to the others with which it is associated within the same containing NewsComponent.

The Role element is empty; its value is represented via the FormalName and associated vocabulary attributes.



At an abstract level, it is not strictly speaking the Role of the NewsComponent itself, but rather the role of the news objects embedded in the NewsComponent in the scope of the container of the NewsComponent. This container is a necessarily a complementary list; in

the case of an equivalents list, all members of that list share the role of their containing component.

It is often useful to discriminate the constituents of a NewsComponent by their role. Roles can show the possible use of an object for a client application. The receiver applications should choose as many constituents of the list as suits their interest, as denoted by the role.

In order to do this, each constituent of the NewsComponent must be itself a NewsComponent containing a Role element. Some NewsComponents are created only to associate a Role to some content.

For example, in a news document that holds two NewsComponents representing photos, the Role of the first image may be 'Thumbnail', and the Role of the second image may be 'Preview'. A NewsComponent representing an article can have a 'Main' Role, and another have a 'Sidebar' Role.

Example 3.3.1:

```
<NewsComponent>
    <NewsComponent>
        <Role FormalName="Thumbnail"/>
        <ContentItem Href="http://www.mycompany.com/content/images/photo1-th.jpg" />
    </NewsComponent>
    <NewsComponent>
        <Role FormalName="Preview"/>
        <ContentItem Href="http://www.mycompany.com/content/images/photo1-pr.jpg" />
    </NewsComponent>
</NewsComponent>
```

The IPTC maintains a set of Role values (e.g. '**Main**', '**Supporting**', '**Thumbnail**', '**Preview**', '**Caption**', '**Abstract**' etc.), along with their descriptions, in a topic set on the IPTC web site. The values of this metadata are intended to be neutral in respect of the media type i.e. appropriate for any media handled as data content, be it text, photo, audio or video. Feedback is welcome from providers, who can create vocabularies that suit to their specific needs.

3.3.2 Equivalence of objects (EquivalentsList)

The **EquivalentsList** attribute is used to state that the constituents of a NewsComponent are alternative representations of the same information; in normal circumstances receivers will only use one of the alternatives supplied. For example, several text parts in different languages are indeed alternative pieces of information, as are several pictures from the same scene in different formats, or alternative articles - one for the Web, one for the WAP - related to the same story. The accepted values of this enumeration are '**yes**' and '**no**', 'no' being the default value which designates a complementary list.

Example 3.3.2:

```
<NewsComponent EquivalentsList="yes">
    <NewsComponent>
        <Role FormalName="Main"/>
        <NewsItemRef NewsItem=" urn:newsml:mycompany.com:20031010:03bg23:1"/>
    </NewsComponent>
    <NewsComponent>
        <Role FormalName="Abstract"/>
        <ContentItem>
            <MediaType FormalName="Text"/>
            <DataContent> ... </DataContent>
        </ContentItem>
    </NewsComponent>
</NewsComponent>
```

3.3.3 Choice between objects (BasisForChoice)

An equivalents list should have a **BasisForChoice** element, which names the property or properties (the criteria) that in the opinion of the provider allows the receiver to make a choice among the alternatives. The selection achieved via the BasisForChoice element should result in one and only one object; if it not the case, the recommended behaviour is to pick the first selected object,

BasisForChoice uses an **XPath** syntax to point to the chosen property (Language, Format ...). The XPath expression is a relative location path and always uses the "child axis" (e.g. shall begin with a './' or './/'). It shall be applied repeatedly, using each of the sibling constituents of the NewsComponent (NewsItem, NewsItemRef, NewsComponent or ContentItem). The first matching result (in document order) in each child represents the information that can be used to choose among the equivalents.

Example 3.3.3:
In this example, the choice is to be made on the Language property of the contained NewsComponents.

```
<NewsComponent EquivalentsList="yes">
    <BasisForChoice>./DescriptiveMetadata/Language/@FormalName</BasisForChoice>
    <NewsComponent>
        <DescriptiveMetadata>
            <Language FormalName="de" />
        </DescriptiveMetadata>
        <ContentItem Href="http://www.mycompany.com/content/de/content-de.xml" />
    </NewsComponent>
    <NewsComponent>
        <DescriptiveMetadata>
            <Language FormalName="es" />
        </DescriptiveMetadata>
        <ContentItem Href="http://www.mycompany.com/content/es/content-es.xml" />
    </NewsComponent>
</NewsComponent>
```

Several BasisForChoice elements may be added; in this case, if several objects are deemed as equals when considering the first basis for choice element, the next basis for choice is tested in order to find the proper object to choose. In that case the optional **Rank** attribute allows providers to place a numerical order on the importance they think should be attached to the different bases for choice. Smaller numbers represent higher importance.

Example 3.3.3a:

In this example, a first filter (BasisForChoice with Rank 1 should be applied on the Language, and a second one (BasisForChoice with Rank 2 on the target audience via OfInterestTo (here we use fictive audiences; 'CEO' - for a business summary and 'Analyst' for a detailed report). The result would be the choice of one NewsComponent amongst the collection.

```
<NewsComponent EquivalentsList="yes">
    <BasisForChoice Rank="1">./DescriptiveMetadata/Language/@FormalName</BasisForChoice>
    <BasisForChoice Rank="2">./ DescriptiveMetadata /OfInterestTo/@FormalName</BasisForChoice>
    <NewsComponent>
        <DescriptiveMetadata>
            <Language FormalName="de" />
            <OfInterestTo FormalName="CEO" />
        </DescriptiveMetadata>
        <ContentItem Href="http://www.mycompany.com/content/de/summary-de.xml" />
    </NewsComponent>
    <NewsComponent>
        <DescriptiveMetadata>
            <Language FormalName="de" />
            <OfInterestTo FormalName="Analyst" />
        </DescriptiveMetadata>
        <ContentItem Href="http://www.mycompany.com/content/de/analysis-de.xml" />
    </NewsComponent>
    <NewsComponent>
        <DescriptiveMetadata>
            <Language FormalName="es" />
            <OfInterestTo FormalName="CEO" />
        </DescriptiveMetadata>
        <ContentItem Href="http://www.mycompany.com/content/es/summary-es.xml" />
    </NewsComponent>
    <NewsComponent>
        <DescriptiveMetadata>
            <Language FormalName="es" />
            <OfInterestTo FormalName="Analyst" />
        </DescriptiveMetadata>
        <ContentItem Href="http://www.mycompany.com/content/es/analysis-es.xml" />
    </NewsComponent>
</NewsComponent>
```

Note: the order of selection can be critical: in the previous example, let us imagine a last NewsComponent having 'en' as a Language and 'CEO' as a value for OfInterestTo: if the (single) 'en' component is selected by a first filter on Language, then no filter of rank 2 is needed. If the two 'Analyst' components are selected by a first filter on OfInterestTo, the second filter on Language may only select the 'de' or 'es' component.

3.3.4 Essential aspect of objects (Essential)

The **Essential** attribute is used to state that the provider considers that this NewsComponent is essential to the meaning of the NewsComponent within which it is contained. This applies to the constituents of complementary lists only, as only one constituent of an equivalents list is to be chosen. The accepted values of this enumeration are '**yes**' and '**no**', 'no' being the default value.

If needed, the "non essential" NewsComponents can be filtered. As an example, a graph which has a specific format, and which is not essential to the news comprehension could be filtered if the recipient has no way to display it; if the graph is tagged as "essential", it is problematic to display the news without it.

If a NewsComponent is marked as essential, it does not mean that its container is essential too, or that all its constituents are essential: when a non essential NewsComponent is used, some of its constituents may then become essential to the

understanding of the information, and other may stay as optional pieces of information. But if a NewsComponent is marked as essential, some of the content of its own constituents has to be usable by all recipients.

Creator: L. Le Meur
Main contributors: J.Rabin, T.Fujiwara, J.Lindgren, N.Onodera

NewsML 1.2 - Guidelines
Chapter 4: Metadata about Content

4 Metadata about content - NewsComponent

In the previous chapter, we described the NewsComponent as a flexible container for news objects. In this chapter we study the NewsComponent as an element holding metadata about content.

4.1 Metadata

A NewsComponent carries metadata that describes its constituents (i.e. ContentItems, NewsComponents or NewsItems).

4.1.1 Classes of metadata

NewsML divides the world of Metadata at the NewsComponent level into four classes:

- **Administrative Metadata** – information about a package of news objects, or about the creation of the content contained in or referenced by the constituents of the NewsComponent.
- **Descriptive Metadata** – information about the content contained in or referenced by the constituents of the NewsComponent.
- **Rights Metadata** – information about the copyrights and usage rights of the content contained in or referenced by the constituents
- **Miscellaneous** – other metadata.

NewsML uses a *controlled vocabulary* mechanism: enumerated values for metadata attributes are not specified in NewsML DTD and schema, but in separate collections of names named *Topic Sets*. The controlled values of metadata elements are usually represented as FormalName attributes associated with optional Scheme and Vocabulary attributes. In the current chapter, the controlled values presentation is simplified, and the use of the *vocabulary attributes* (Scheme and Vocabulary) is not described.

> **Note about the location of IPTC Topic Sets:** The IPTC maintains a set of topic sets, along with their descriptions in different languages. The IPTC Web site has a specific area where users can view and download the IPTC controlled vocabularies.
>
> Check the IPTC web site for updates: **http://www.iptc.org/metadata**
>
> **Going deeper:** detailed information about controlled vocabularies and TopicSets is given in **Chapter 8**, "**NewsML controlled vocabularies**".
>
> **Note about NewsML metadata vs. Dublin Core metadata:**
> Dublin Core represents an alternative scheme for the classification of printed documents. It was intended to be simpler than the USA librarian's coding scheme, but has become more complex over time. Some of the Dublin Core is relevant to the NewsML domain, but it is not easy to apply. During NewsML development, it was considered advantageous to have mapping to Dublin Core but not to be constrained by the Dublin Core constructs, as the Dublin Core was developed from the bibliographic domain and not from a news production viewpoint.

Several metadata properties chosen by the IPTC have Dublin Core counterparts. The equivalence between IPTC NewsML metadata and Dublin Core metadata is described in this chapter as specific notes.

4.1.2 Metadata inheritance

A NewsComponent carries metadata that describes its constituents. As metadata attached to the NewsComponent is inherited by its constituents, it should be attached to the topmost applicable NewsComponent (i.e. the NewsComponent element nearest to the root of the XML document) to minimise duplication of that metadata further away from the root. Constituents of NewsComponents may override metadata applied at a higher level.

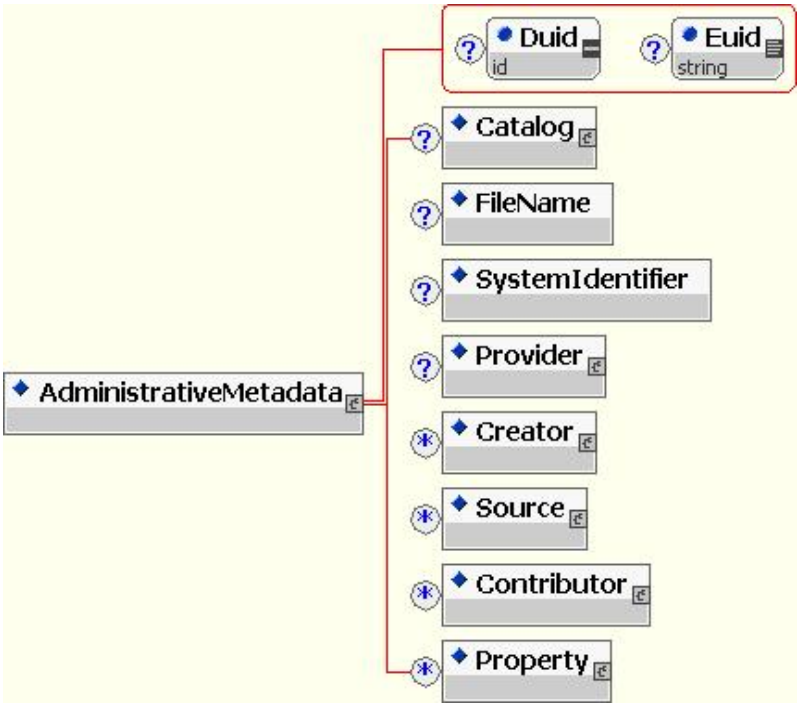
Note: There is no way to force a constituent of a NewsComponent to adopt metadata defined at the NewsComponent's level. For example, a NewsComponent cannot supersede metadata defined in a child NewsItem, a feature that could be interesting when the NewsItem is represented via its reference and cannot be modified.

Open question: When a NewsItem is embedded in a NewsComponent, and this NewsComponent supports some metadata, should the metadata inheritance traverse the NewsItem boundaries? This issue will be studied in the scope of NewsMLv2.

4.2 Administrative Metadata

Administrative metadata are intended to provide information about a **package of news objects, i.e. a NewsItem**; some of those elements also provide information about the **creation** of the content contained in or referenced by the constituents of a NewsComponent.

The two first metadata elements describe "where" the information might be found; the other elements describe "who" created the information.



A common set of metadata is defined by the IPTC; other specific metadata may be defined by a provider via the Property element.

> **Going deeper:** detailed information about the Property element is given in the **Chapter 9,** "**Extension mechanisms**".

4.2.1 FileName and SystemIdentifier

The optional **FileName** element identifies the suggested or actual storage file name for a NewsItem (or NewsML instance, that is a full <NewsML>… </NewsML> tree) stored in a file system.

The optional **SystemIdentifier** element specifies a system address (such as an http URL pointing at a file or a dynamic web page) where the NewsItem or NewsML instance can be found. This is a system identifier for the remote resource, in the sense defined by the XML 1.0 Specification.

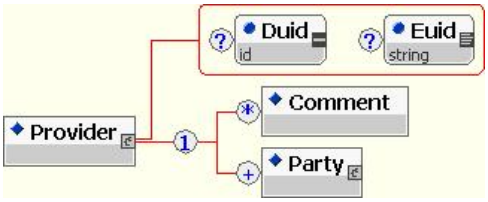
Be aware that FileName and SystemIdentifier directly refer to the parent NewsItem or NewsML instance. Thus, those elements should only be included in the "main" NewsComponent of a NewsItem (that is the required NewsComponent directly included in a NewsItem); for more details, see **Chapter 5,** "**The management level – NewsItem**" and the paragraph about "**NewsItem and storage**".

Example 4.2.1:

```
<NewsItem>
    <Identification> ... </Identification>
    <NewsManagement> ... </NewsManagement>
    <NewsComponent>
        <AdministrativeMetadata>
            <FileName>NewsMLStory.xml</FileName>
            <SystemIdentifier>http://ww.mycompany.com/stories/NewsMLStory.xml</SystemIdentifier>
        </AdministrativeMetadata>
    </NewsComponent>
</NewsItem>
```

4.2.2 Provider

The optional **Provider** element identifies the company or organization (or an individual) that released a NewsItem and made it available for publishing.



The mandatory **Party** element is used to provide information about the Provider. A party identifier is mandatory, using the FormalName and associated vocabulary attributes. More information about the Party element is given in chapter 4.2.5.

The IPTC maintains a set of Provider values, along with their descriptions in different languages.

Be aware that Provider directly refers to its parent NewsItem. Thus, this element should only be included in the "main" NewsComponent of a NewsItem.

Dublin Core: Provider has the same semantics as the Dublin Core '**publisher**' property; the DC definition of 'publisher' is "The entity responsible for making the resource available."

Example 4.2.2:

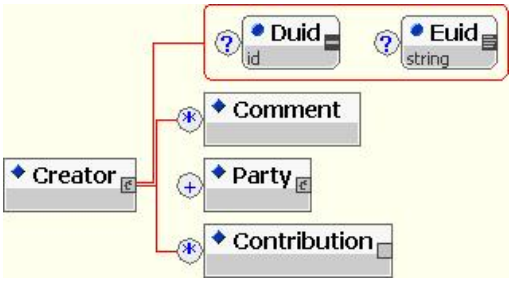
```
<NewsItem>
    <Identification> ... </Identification>
    <NewsManagement> ... </NewsManagement>
    <NewsComponent>
        <AdministrativeMetadata>
            <Provider>
                <Party FormalName="MyCompany" />
            </Provider>
        </AdministrativeMetadata>
    </NewsComponent>
</NewsItem>
```

4.2.3 Creator and Contributor

Some elements may refer to the NewsItem as well as its children objects; this is the case for Creator and Contributor.

Found in the main NewsComponent of a NewsItem, the optional and repeatable **Creator** element identifies the party that created the NewsItem. Found in any NewsComponent (including the main NewsComponent of a NewsItem), it identifies the party that created the content of the constituents of this NewsComponent. If some children of the NewsItem do not share some aspects of that metadata, then it is necessary to introduce a child NewsComponent with AdministrativeMetadata that overrides the values set at the NewsItem level.

Multiple Creators are allowed from NewsMLv1.2, and used when the constituents are a team creation.



The optional and repeatable **Contributor** element identifies an individual or organization that modified or enhanced the NewsItem or the constituents of a NewsComponent after their creation.

Dublin Core: Creator and Contributor have the same semantics as their Dublin Core counterparts. In Dublin Core 'creator' is defined as "an entity primarily responsible for making the content of the resource," and a contributor as "an entity responsible for making contributions to the content of the resource".

Creator and Contributor have an identical structure:

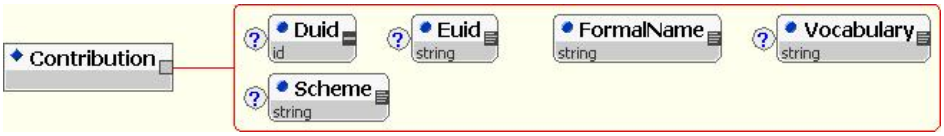
The optional and repeatable **Comment** element is used to provide some relevant free text information about this creator or contributor. A language can be set and a Comment type can be added via the FormalName attribute.

Going deeper:
More information about the Comment element is given in **Chapter 2.5** "**The Comment element**".

The mandatory **Party** element is used to provide information about the Creator or Contributor. A party identifier is mandatory, using the FormalName and associated vocabulary attributes; but the names of the people involved in the editing of the story is usually internal information to the provider, and no TopicSet is published for it. It is thus recognized that the Party/@FormalName attribute can be a literal value that corresponds to a specific provider defined vocabulary. See chapter

Warning: Even if Party is repeatable in Creator and Contributor, its repetition is not recommended, as multiple Creator and Contributor elements are seen as a more adequate representation.

The optional and repeatable **Contribution** element was created in NewsMLv1.2 in order to represent the contribution of the Creator or Contributor as for the authoring of a NewsComponent. The Contribution element is empty, and the value is given via the FormalName and associated vocabulary attributes.



The IPTC has not defined yet a formal controlled vocabulary for this element, but example values are '**Reporter**', '**Writer**', '**Editor**', '**Photographer**', '**CaptionWriter**', '**Translator**'.

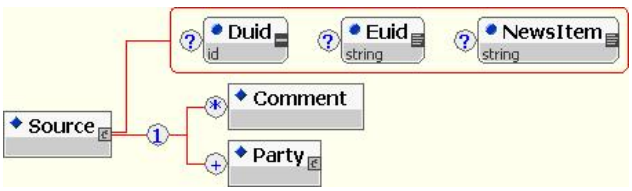
Example:

```
<NewsComponent>
    <AdministrativeMetadata>
        <Creator>
            <Party FormalName="john.doe@mycompany.com"
            Vocabulary="urn:newsml:mycompany.com:20031010:people"/>
            <Contribution FormalName="Writer"/>
        </Creator>
        <Contributor>
            <Comment xml:lang="fr">Traduction française</Comment>
            <Party FormalName="Julien Dupond" />
            <Contribution FormalName="Translator"/>
        </Contributor>
    </AdministrativeMetadata>
</NewsComponent>
```

In this sample, the writer is described via its identifier in the people listing of 'mycompany.com'. The French translator (as the Contribution and Comment state) is identified by its full name.

4.2.4 Source

The optional and repeatable **Source** element is defined as "An individual and/or company or organisation that provided source material for a news object".



The mandatory **Party** element is used to provide information about the Source. A party identifier is mandatory, using the FormalName and associated vocabulary attributes. See chapter 4.2.5.

Found in the main NewsComponent of a NewsItem and having a **NewsItem** attribute, it identifies the party that initially provided the content of the NewsItem in a syndication chain; the NewsItem attribute then provides the URN of the NewsItem that is being syndicated. Found in any NewsComponent with no NewsItem attribute, it identifies the party that provided source material for the content used in the constituents of this NewsComponent.

In a syndication model, a sequence of Source elements is used to indicate the sequence of syndicators through which a NewsComponent has passed. The first Source party appearing in the collection was the first syndicator in the chain.

Again, a **Comment** can provide any additional relevant information.

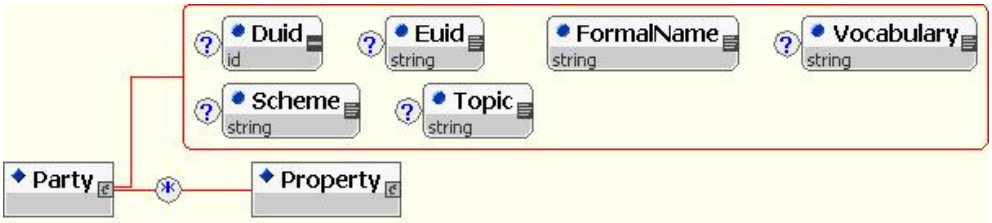
> **Dublin Core:** Source does not have exactly the same meaning as its Dublin Core counterpart; the 'source' Dublin Core property is "a reference to a resource from which the present resource is derived". Even if the NewsItem attribute plays a similar role, the equivalent NewsML metadata is named DerivedFrom, and its value is a NewsItem identifier; it is present at the news management level.

Example 4.2.4:

```
<NewsItem>
    <Identification> ... </Identification>
    <NewsManagement> ... </NewsManagement>
    <NewsComponent>
        <AdministrativeMetadata>
            <Source>
              <Party FormalName="ACompany" NewsItem="urn:newsml:acompany:20031010:NewsMLStory"/>
            </Source>
        </AdministrativeMetadata>
    </NewsComponent>
</NewsItem>
```

4.2.5 The Party element

The **Party** element is used to provide information about the Source, Provider, Creator or Contributor elements.



Detailed information about a Party may be given using the Property sub-elements, using in such a case a vCard-like structure.

> **Note about IPTC and vCard:** During June 2003 IPTC meeting, it was agreed that IPTC would take part in an initiative at OASIS to create an XML version of vCard.

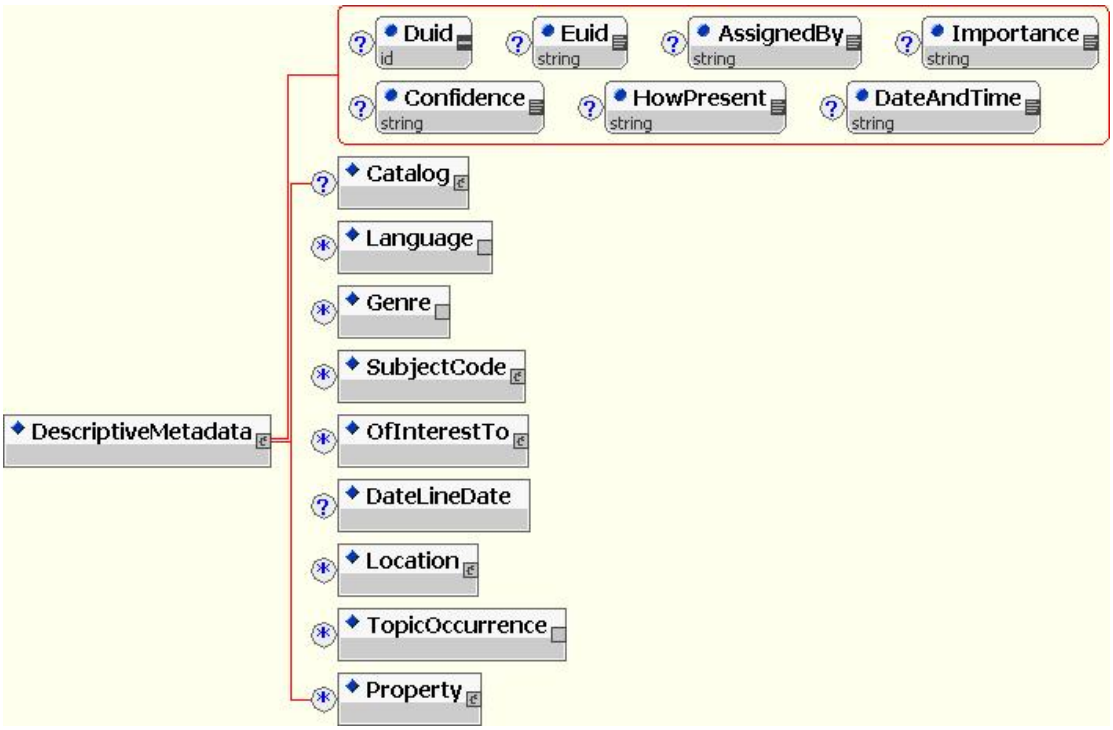
> **Note about feedback response:**
> As a particular case of the use of this feature, a basic level of feedback response tag can be set via the eMail property of the vCard structure.

Example 4.2.5:

```
<NewsComponent>
    <AdministrativeMetadata>
        <Creator>
            <Party FormalName="jdoe">
                <Property FormalName="vCard">
                    <Property FormalName="FN" Value="John Doe"/>
                    <Property FormalName="Email" Value="john.doe@mycompany.com"/>
                    <Property FormalName="Tel" Value="+44 (20) 4455 6677"/>
                </Property>
            </Party>
            <Contribution FormalName="Writer"/>
        </Creator>
        <Contributor>
            <Comment xml:lang="fr">Traduction française</Comment>
            <Party FormalName="jdupond">
                <Property FormalName="vCard">
                    <Property FormalName="FN" Value="Julien Dupond"/>
                    <Property FormalName="Email" Value="jules.dupond@masociete.com"/>
                    <Property FormalName="Tel" Value="+33 0144556677"/>
                </Property>
            </Party>
            <Contribution FormalName="Translator"/>
        </Contributor>
    </AdministrativeMetadata>
</NewsComponent>
```

4.3 Descriptive Metadata

Descriptive metadata provides information about the content contained in or referenced by the constituents of a NewsComponent.

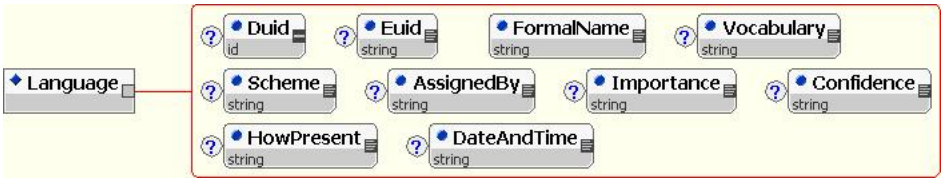


A common set of metadata is defined by the IPTC, and other specific metadata may be defined by a provider using the Property element.

> **Going deeper:** detailed information about the Property element is given in the **Chapter 9,** "**Extension mechanisms**".

4.3.1 Language

The optional and repeatable Language element indicates the language used in the constituents of a NewsComponent. More than one language can be specified.



> **Dublin Core:** Language has the same semantics as its Dublin Core counterpart; "A language of the intellectual content of the resource".

The full set of language tags used on the Internet is specified by **RFC 3066** (e.g. 'en-US', 'fr-BE' or simply 'de'), and keeps evolving at the pace of ISO work. The RFC 3066 values are already used for xml:lang attributes in every XML document.

Maintaining a fixed Topic Set for this massive combination of values would be a daunting task. So the IPTC choose in October 2003 to adopt the following rule for NewsML1.2 documents: **The FormalName attribute of the Language element is controlled by RFC 3066 and not by an IPTC topic set**.

> **Note concerning the Scheme and Vocabulary attributes:**
> As no Topic Set is used for the control of the Language element, the Scheme and Vocabulary attributes are not present in this element.

> **Note concerning previous versions of NewsML:**
> The IPTC originally created a set of **ISO 639** values (e.g. 'en', 'fr'), along with their descriptions, in the "**topicset.iso-language.xml**" topic set. This topic set is still usable for NewsML1.0 and NewsML1.1 documents.

> **Going deeper:**
> Detailed explanation of use of the Language element is given in IPTC document NMLS 0308, appended to this document.

4.3.2 Genre

The optional and repeatable **Genre** element indicates the "style of expression" used in the constituents of a NewsComponent. The Genre element is empty; the proper genre is represented via the FormalName and associated vocabulary attributes.

The IPTC maintains a set of Genre values (e.g. '**Current**', '**Analysis**', '**Feature**', '**Obituary**', '**Profile**', '**Interview**' etc.), along with their descriptions in different languages.

The values of this metadata should cover any media handled as data content, be it text, photo, audio or video.

> **Dublin Core:** Genre has the same semantics as the Dublin Core 'type' property; "The nature or genre of the content of the resource".

Default value: if no Genre is included, the default value is 'Current'.

4.3.3 SubjectCode

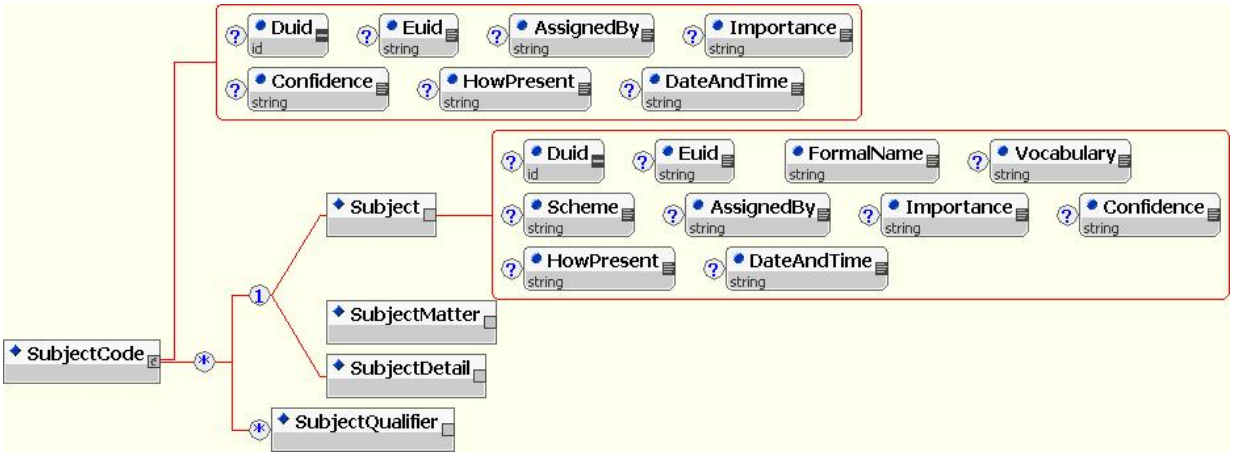
The optional and repeatable **SubjectCode** element indicates the subject or news category associated with the constituents of a NewsComponent. The SubjectCode element is a structure, and contains a Subject, SubjectMatter or SubjectDetail element; it also supports an optional and repeatable SubjectQualifier element.

The Subject, SubjectMatter and SubjectDetail elements represent a three-level taxonomy, mapped from the structure of the IPTC SRS (Subject Reference System), a three-level rich set of concepts targeted to news description. A Subject element is used when choosing a value from the first level of the taxonomy, a SubjectMatter corresponds to the second level, a SubjectDetail to the third; the proper subject value is represented via the FormalName and associated vocabulary attributes of the selected element.

It is not necessary to include Subject or SubjectMatter if a SubjectDetail is encoded nor a Subject if a SubjectMatter is encoded.

The SubjectQualifier element amplifies and adds information to the subject, especially in the sports area.

Subject, SubjectMatter, SubjectDetail and SubjectQualifier have all the same structure.



Dublin Core: SubjectCode has the same semantics as the Dublin Core 'subject' property; "The topic of the content of the resource".

Example 4.3.3:

```
<DescriptiveMetadata>
    <SubjectCode>
        <!-- sport, swimming, 50 m butterfly, final, men-->
        <Subject FormalName="1506206"/>
        <SubjectQualifier FormalName="15000001"/>
        <SubjectQualifier FormalName="15000024"/>
    </SubjectCode>
    <SubjectCode>
        <!— applied science -->
        <Subject FormalName="13001000"/>
    </SubjectCode>
</DescriptiveMetadata>
```

The IPTC SRS evolves at a steady pace, and currently includes more than 1200 terms.

The IPTC maintains a set of Subject and SubjectQualifier values, along with their descriptions in different languages. These topic sets are published on the IPTC Web site.

4.3.3.1 Use of generic subject taxonomies

When the IPTC taxonomy have the precision needed by the provider, any specialized taxonomy can be used as a complement or as an alternative. In this case it is recommended to use the Subject element as a support for the specific subject value. Care should be taken when this feature is used, as the normal choice – the IPTC SRS -- is a great way to enhance categorisation compatibility in the news area.

Example 4.3.3.1:

```
<DescriptiveMetadata>
    <SubjectCode>
        <Subject FormalName="I" Vocabulary="urn:newsml:afp.com:20011001:AFPCatCodes:1"/>
    </SubjectCode>
</DescriptiveMetadata>
```

4.3.3.2 Recommended structure for multiple subjects

Even if the SubjectCode structure allows for a repetition of the group [(Subject|SubjectMatter|SubjectDetail), SubjectQualifier*] the use of this feature is not recommended, as this may lead to confusion and difficulties in parsing and processing the subject information. The simplest way to reduce the potential for confusion would be to remove the capability for multiple appearances of the SubjectCode child elements. However, this would break existing valid implementations that use the multiple child elements. This will not be done at present to preserve backward compatibility.

> **Recommendation:** The recommended approach is to only encapsulate one set of child elements within each SubjectCode. This is particularly useful where SubjectQualifiers are included since they will only refer to their immediate sibling

Example 4.3.3.2: Recommended structure for multiple subjects:

```
<DescriptiveMetadata>
    <SubjectCode>
        <Subject FormalName="15000000"/>
        <SubjectQualifier FormalName="15000010"/>
        <SubjectQualifier FormalName="15000006"/>
    </SubjectCode>
    <SubjectCode>
        <SubjectMatter FormalName="15054000"/>
        <SubjectQualifier FormalName="15000023"/>
    </SubjectCode>
</DescriptiveMetadata>
```

4.3.3.3 SubjectCode FormalName Decoding

The FormalName value is a unique eight-digit number that is assigned to each entry in the three-level Subject hierarchy. This number is broken down as follows:

- The first two digits indicate the top-level Subject. The valid values are 01 through 17. (Leading zero is mandatory).
- The next three digits indicate the SubjectMatter when read in conjunction with the parent Subject number. Default is 000, used when no SubjectMatter is specified. The rest of the values (001-999) must be used in conjunction a two-digit Subject number.
- The last three digits, when preceded by valid Subject and SubjectMatter numbers, indicate SubjectDetail. The last three digits separately are used to indicate a SubjectQualifier – but only in concert with a Subject number. The middle three digits must be 000 when providing a SubjectQualifier.

The appropriate Subject, SubjectMatter and SubjectDetail data may be extracted (assuming they are validly encoded) from the FormalName as follows:

Examples 4.3.3.3:

Subject

	Subject	SubjectMatter	SubjectDetail
13000000	13xxxxxx	13000xxx	13000000
	Science and technology	(none)	(none)

SubjectMatter

	Subject	SubjectMatter	SubjectDetail
07002000	07xxxxxx	17002xxx	17002000
	(health)	epidemic and plague	(none)

SubjectDetail

	Subject	SubjectMatter	SubjectDetail
04015002	04xxxxxx	04015xxx	04015002
	(economy, business and finance)	(transport)	Railway

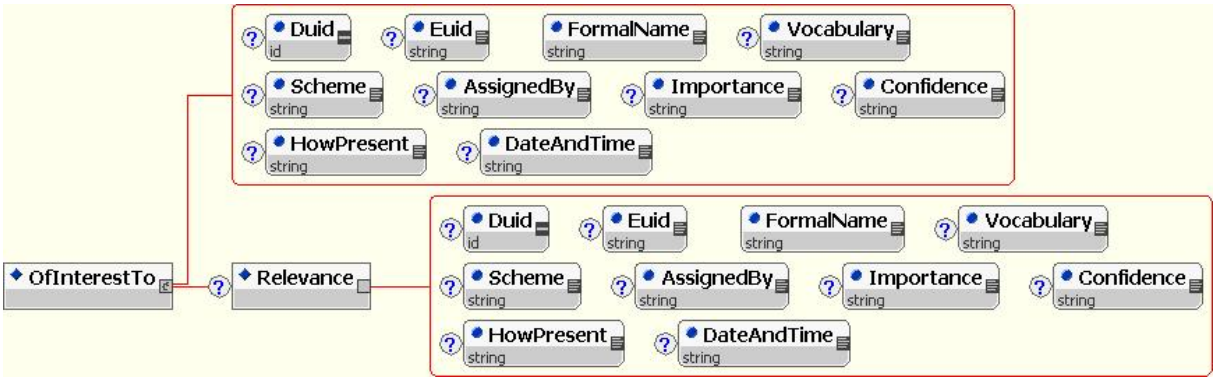
SubjectQualifier

	Subject	SubjectMatter	SubjectQualifier
15000001	15xxxxxx	15000xxx	15000001
	(sport)	(none)	Men

The numbering scheme allows for logical extensions to all of these lists. There is no relevance in the sequence of numbers allocated to entries in the scheme. They were derived initially from an alphabetical sort of the original lists and do not contain an implied hierarchy. However, the first two digits are always the same for a given Subject and the digits at positions 3, 4 and 5 are the same for a given SubjectMatter within a Subject.

4.3.4 OfInterestTo (Target audience)

The **OfInterestTo** element indicates the target audience of the constituents of a NewsComponent. Its **Relevance** sub-element indicates the relevance of a NewsItem to a given target audience. The proper audience and relevance values are represented via the FormalName and associated vocabulary attributes of each element.



The IPTC maintains an experimental set of OfInterestTo values (e.g. '**General**', '**Over50**', '**Parents**', etc.), along with their descriptions in different languages. This topic set is published on the IPTC Web site. A set of Relevance values (e.g. '**High**', '**Medium**', '**Low**') is also maintained by the IPTC. Providers can create vocabularies fitting to their specific needs.

4.3.5 DateLineDate and Location of origin of the news

The DateLineDate and Location elements have been added to DescriptiveMetadata in NewsMLv1.1. They are relative to the creation of news, i.e. the date a story was written, the location it was filed from, or the date and location a picture was shot.

The DateLineDate element may be used to provide a logical equivalent for the date of origin of the news object, as found in the news agency "dateline". Its content is in the ISO8601 Basic Date Format.

The Location element may be used to provide a logical equivalent for the location of origin of the news object, as found in the news agency "dateline". In this case the HowPresent attribute must be set to '**Origin**'. The Location content model consists of the Property element that may occur zero or more times.

The IPTC maintains a set of Location values (e.g. '**Country**', '**CountryArea**', '**City**', '**SubLocation**', '**WorldRegion**'), along with their descriptions in different languages.

> **Rule:** The use of the iptc-location topicset is **normative** in the news community (i.e. news providers *must* use it and no other vocabulary).

Example 4.3.5:

```
<DescriptiveMetadata>
    <DateLineDate>20031010</DateLineDate>
    <Location HowPresent="Origin">
        <Property FormalName="Country" Value="US"/>
        <Property FormalName="CountryArea" Value="DC"/>
        <Property FormalName="City" Value="Washington"/>
        <Property FormalName="SubLocation" Value="The White House"/>
        <Property FormalName="WorldRegion" Value="North America"/>
    </Location>
</DescriptiveMetadata>
```

Although the Location element allows for arbitrary nesting of their Property sub-elements, it is not recommended that Property elements are nested unless it is considered absolutely essential to avoid ambiguity.

4.3.6 Location

Beyond the '**Origin**' location (just seen above), the structure of the Location element allows the formal representation of any Location that appears in the constituents of a NewsComponent. The value of the **HowPresent** attribute should be set accordingly; two current values of the HowPresent TopicSet are of special use:

- '**RelatesTo**': The constituents of the NewsComponent have a reference to the location.
- '**Event**': The constituents of the NewsComponent refer to an event that took place at this location.

> **Note:** Be careful to distinguish the location relative to the creation of news (see 4.3.5) and the locations relative to the news itself described here.

4.3.7 TopicOccurrence (Topic in the news)

The **TopicOccurrence** element indicates that a Topic occurs in the constituents of a NewsComponent.

The optional **HowPresent** attribute indicates the nature of their occurrence. The value of the **Topic** attribute must consist of a fragment identifier, i.e. a # character followed by the value of the Duid attribute of a Topic in the current document (i.e. a value in a local topicset). So the value can not be a URL pointing at the topic in a remote TopicSet.

Going deeper: detailed information about Topics and TopicSets is given in the **Chapter 8, "NewsML controlled vocabularies"**.

Example: 4.3.7

A paper related to the murder of Cdr. Massoud in Afghanistan can be enriched this way:

```
<NewsComponent>
    <TopicSet Duid="LocalTopicSet">
        <Topic Duid="topic1" Details="http://www.afghan-info.com/Politics/Massoud/Biography.htm">
            <!-- Details given in a web page -->
            <!-- extracted from urn:newsml:afp.com:20020701:topicset.AfghanPeople:3 -->
            <TopicType FormalName="Person" />
            <FormalName>Ahmed Shah Massoud</FormalName>
            <Description xml:lang="en-US">Cdr Massoud</Description>
            <Property FormalName="Nationality" Value="AFG"
ValueVocabulary="urn:newsml:iptc.org:20001006:topicset.iso-country:3" ValueScheme="ISO3166-alpha3"/>
        </Topic>
        <Topic Duid="topic2">
            <!-- extracted from urn:newsml:afp.com:20001006:topicset.IslamistOrganizations:5 -->
            <TopicType FormalName="Organization" />
            <FormalName>Al Qaeda</FormalName>
        </Topic>
    </TopicSet>
    <DescriptiveMetadata>
        <TopicOccurrence Topic="#topic1" HowPresent="Prominent" />
        <TopicOccurrence Topic="#topic2" HowPresent="RelatesTo" />
    </DescriptiveMetadata>
</NewsComponent>
```

4.3.8 Assignment attributes

The DescriptiveMetadata element and its sub-elements all support several assignment attributes:

- **AssignedBy**: An indication of who, or what system, assigned the current metadata (the element that supports this attribute).
- **Importance**: A rating of the importance the party assigning the current metadata attaches to it. The IPTC maintains a set of values (e.g. 'High, 'Medium', 'Low'), along with their descriptions in different languages.
- **Confidence**: A rating of the confidence with which the current metadata was assigned. The IPTC maintains a set of values (e.g. 'Full', 'High', 'Medium', 'Low'), along with their descriptions in different languages
- **HowPresent**: An indication of the way in which the current metadata applies. The IPTC maintains a set of values (e.g. 'Prominent'), along with their descriptions in different languages.
- **DateAndTime**: The date and (optionally) time at which the current metadata was assigned.

More detailed information is given in the NewsML Functional Specifications.

Note: The assignment attributes as described in this paragraph are also supported by RightsMetadata sub-elements and extended metadata (the Property element).

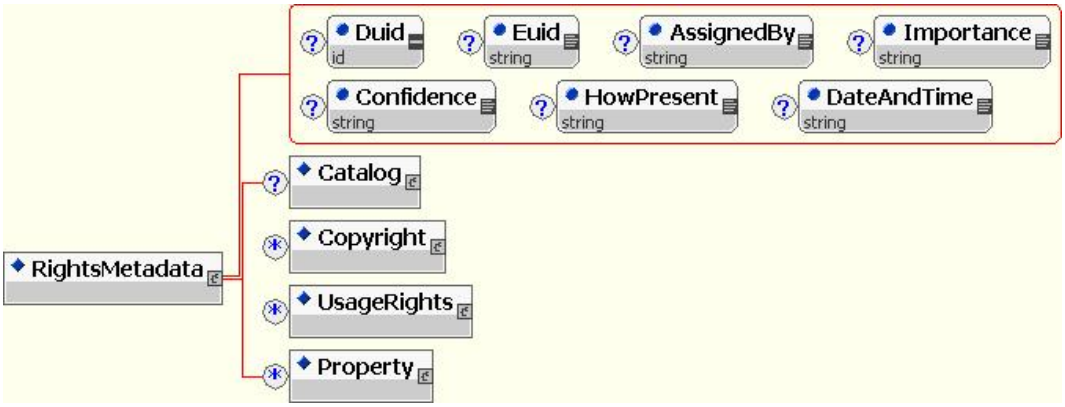
Example 4.3.8:

```
    <DescriptiveMetadata AssignedBy="John Doe" Confidence="High" DateAndTime="20031010T100000+0100">
        <SubjectCode Importance="High"><Subject FormalName="01000000"/></SubjectCode>
        <SubjectCode Importance="Low"><SubjectMatter FormalName="03001000"/></SubjectCode>
        <Location HowPresent="Origin"><Property FormalName="Country" Value="FRA"/></Location>
    </DescriptiveMetadata>
```

4.4 Rights Metadata

The RightsMetadata element contains information about the rights pertaining to the constituents of a NewsComponent, and any relevant usage rights that have been granted by the copyright holder to other parties.

Two sets of metadata are defined: **copyrights** and **usage rights**.



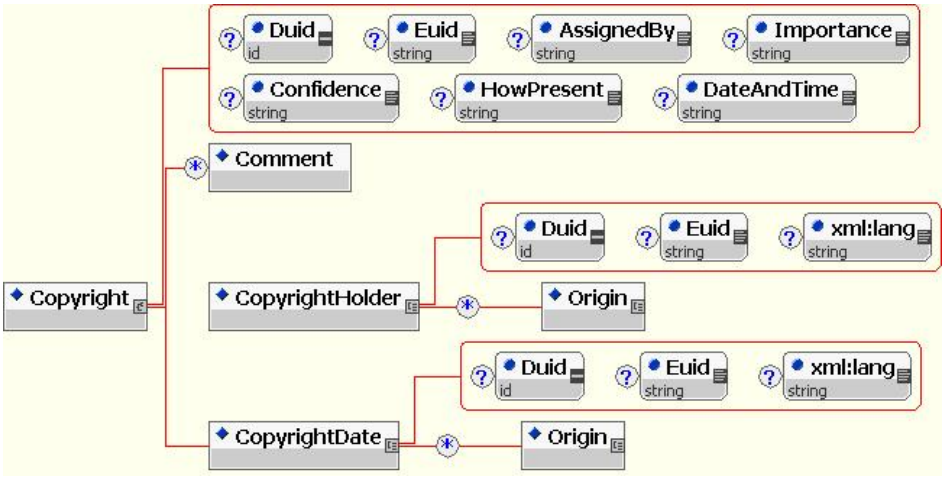
Rights metadata values are not controlled by vocabularies like administrative and descriptive metadata, but are instead extended text strings.

Extended strings are mixed elements and may contain '**Origin**' elements. The Origin element is a wrapper for all or part of the text, which provides a pointer to machine-interpretable data corresponding formally to what is being described here in natural language.

> **Going deeper:** examples of the use of the origin element are given in 4.5.
>
> **Note about the evolution of NewsML:** the IPTC will evaluate the main rights metadata XML standards (especially XrML and ODRL) during the next major evolution of NewsML.

4.4.1 Copyright

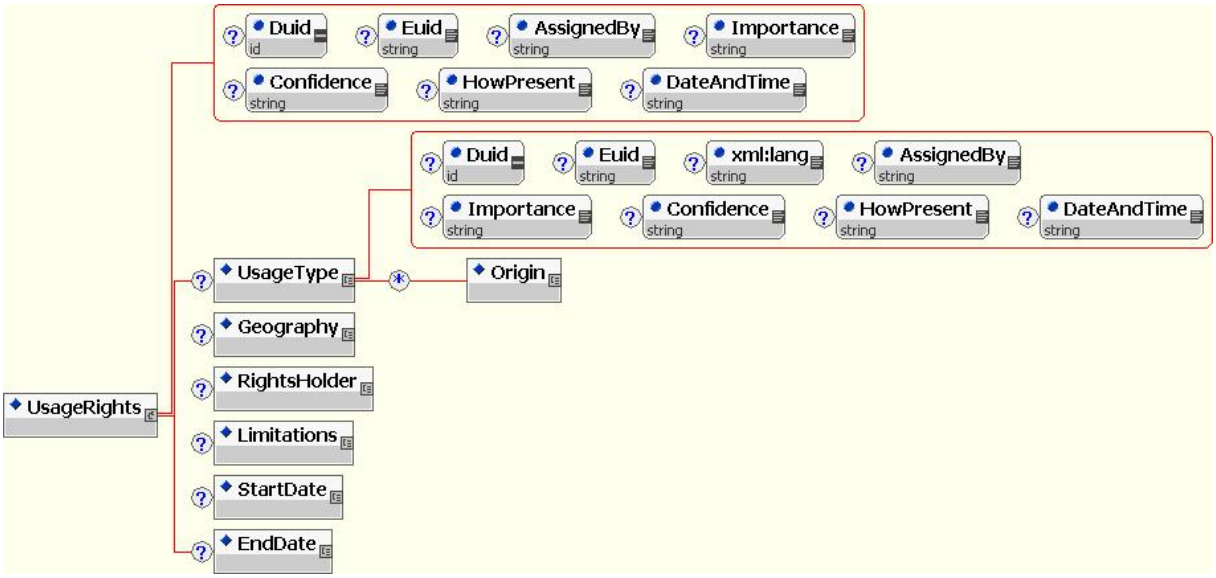
The **CopyrightDate** and **CopyrightHolder** elements provide natural-language statements of the copyright date and ownership.



4.4.2 Usage rights

The UsageRights set of metadata provides information about the usage rights pertaining to the constituents of a NewsComponent.

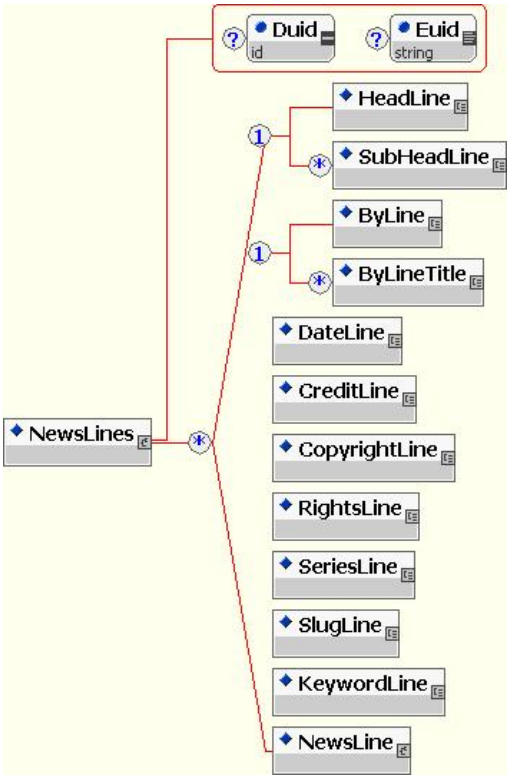
- **UsageType** provides a natural-language indication of the type of usage to which the rights apply;
- **Geography** indicates the geographical area or areas to which specified usage rights pertain;
- **RightsHolder** indicates who has the usage rights;
- **Limitations** indicates any restrictions on the use of the content of the NewsComponent;
- **StartDate** and **EndDate**, indicate the time period over which the stated rights apply.



4.5 NewsLines

NewsLines expose aspects of news as natural-language labels. They represent characteristic properties of news objects such as headline, byline, dateline or copyright information, and are typically displayed alongside the content of an object or in place of the object in a list of such objects, providing a means of selection among them.

The order of NewsLines is free, following the practice of each provider.



Newslines are distinct from content, as they were originally designed to provide (among other things) a human-readable representation of parts of the metadata accompanying content. As such they can be considered as a “card index”, especially useful for content syndication when the content is referenced and not included in-line. A clear example is a binary file (e.g. a picture), for which the headline is a kind of “catch line”.

For most media types, the content part of the news object can also logically include similar information (e.g. as NITF or xhtml headline or sub-headline). Equally, many providers are currently using NewsLines to contain information - such as headline, byline, dateline - which is not represented anywhere else in the document

For example, if NITF is used, a story title is represented as a NewsML HeadLine and the nitf/head/title element is absent.

This has the advantage of avoiding any duplication of data, but the provider sacrifices the possibility to mark the label as flexibly as with real content. At present NewsML does not provide a mechanism to allow receivers to determine whether NewsLine information is duplicated in ContentItems or not.

> **Note about the evolution of NewsLines:**
> A standardization of the use of NewsLines will be studied in depth during the next major evolution of NewsML.

A common set of NewsLines is defined by the IPTC, and other specific NewsLines may by defined by a provider using a generic element.

The **HeadLine** element provides a displayable headline and the **SubHeadLine** element provides a multiple subsidiary headline.

The **ByLine** element provides a natural-language statement of the author/creator information and the **ByLineTitle** element provides a natural-language statement of the title of the author/creator information.

The **DateLine** element provides a natural-language statement of the date and/or place of the news object's creation. Traditionally a dateline indicates when and where an item is created, not necessarily the time and place of the content. As an example a dateline *Paris Aug 8 (UPI)* could head a story about crime in the Ivory Coast, because the story was actually written in Paris. Also, by tradition a dateline will follow the stylebook of the information provider and possibly leave out certain time and location information that could be useful for specifying searches of a database. For example there are 15 cities in the U.S.A. with the name of Paris.

The **CreditLine** element provides a natural-language statement of credit information.

The **CopyrightLine** element provides a natural-language statement of the copyright information.

The **RightsLine** element provides a displayable version of rights information. Note that this is distinct from copyright information. Copyright information is about who owns a news object; rights information is about who is allowed to use it, in what way and under what circumstances.

The **SeriesLine** element provides a displayable version of information about a news object's place in a series.

The **SlugLine** element provides a string of text used to display a news object's slug line (note that the meaning of the term "slug line", and the uses to which it is put, are a matter for individual providers to define).

The **KeywordLine** element provides a displayable set of keywords relevant to a news object.

NewsLine elements allow for the inclusion of a type of news line not included in the NewsML specification.

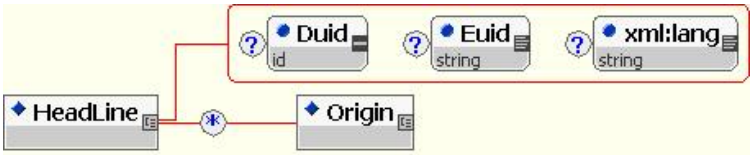
> **Going deeper:** detailed information about the NewsLine element is given in the **Chapter 9,** "**Extension mechanisms**".

Example 4.5:

```
<NewsLines>
    <HeadLine>Firefighters battle to save towns as California fires kill 2 more</HeadLine>
    <SubHeadLine>by Marc Lavine</SubHeadLine>
    <SubHeadLine>= (PICTURES + GRAPHICS) =</SubHeadLine>
    <NewsLine>
        <NewsLineType FormalName="AdvisoryLine"/>
        <NewsLineText>ATTENTION - UPDATES ///</NewsLineText>
    </NewsLine>
    <DateLine>LOS ANGELES, Oct 29 (AFP) -</DateLine>
</NewsLines>
```

4.5.1 Structure of a NewsLine

All NewsLine elements share the same structure:



4.5.1.1 The xml:lang attribute

NewsLines are inherently textual and may be present in multiple languages. The "xml:lang" attribute is used to specify the language used in a NewsLine element.

> **Going deeper:** a detailed description of the way in which multiple language news can be represented is found in IPTC document NMLS 0308 (appended to the document) "**The expert zone / Multilingual news**".

4.5.1.2 The Origin marker

A provider can mark topics of interest with **Origin**. Origin is a wrapper for a piece of text, which provides a pointer to an item of data corresponding formally to what is being described here in natural language. This is a back-link from the NewsLine to the metadata it represents in a human readable way.

> **Note:** the Origin element is also found in RightsMetadata elements.

The **Href** attribute on the Origin element identifies the relevant data, usually as a fragment identifier identifying the Duid of an element in the current document.

As an example, the ByLine may contain an Origin element linking the fragment which contains the byline name to the Creator metadata; the DateLine may contain an Origin element linking the date to the DateLineDate metadata, and the location to the Location metadata.

One of the most common uses of the Origin element is as an in-situ marker for Topics. As an example, the HeadLine may contain an Origin element linking a fragment which contains a people's name to a Topic element defined in a local topic set, which gives more information on this named entity.

Example 4.5.1.2:
These NewsLines could be associated with the paper related to the murder of Cdr. Massoud in Afghanistan, as introduced in the previous paragraph about Topic extraction:

```
<NewsLines>
    <HeadLine><Origin Href="#topic1">Crd Massoud</Origin> murdered at he's headquarters</HeadLine>
    <SubHeadLine><Origin Href="#topic2">Al Qaeda</Origin> suspected of the crime</SubHeadLine>
    <DateLine><Origin Href="#location">Kabul</Origin>, Sept 9 (MYCOMPANY)</DateLine>
</NewsLines>
```

> **Note about Href:**
> Reading the NewsML specifications, it appears that Href can also be an http, ftp of other type of URL, or a NewsML URN optionally followed by a fragment identifier. The use of such an URL is not recommended, as no standard use of such a hyperlink has been approved by IPTC. The use of a NewsML URN followed by a fragment identifier (e.g. urn:newsml:iptc.org:20001006:topicset.iso-country:4#isoc250 for France) is not recommended either, as it conflicts with the current standard use of controlled vocabularies (i.e. use of FormalName, Vocabulary and Scheme attributes).

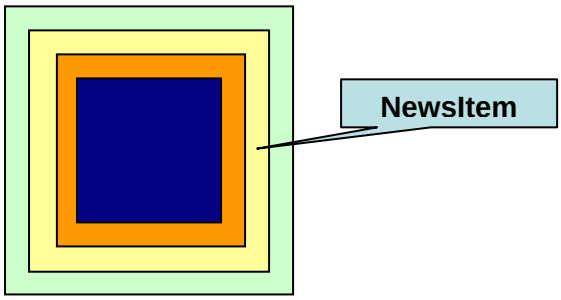
Creator: L. Le Meur
Main contributors: J.Rabin, T.Fujiwara, J.Lindgren, N.Onodera, H.Shinotsuka

NewsML 1.2 - Guidelines
Chapter 5: The Management Level

5 The management level – NewsItem

5.1 Context of the NewsItem

Above the structure level, the NewsItem is the prime unit of news management in NewsML.



A NewsItem is an *identified* and *publishable* piece of news. This is the news object that a provider will create, store, manage, reuse, link *to* and *from* other NewsItems; this is the unit of interchange in a news environment, a potential entry point into a web of NewsItems that reference or include each other.

A NewsItem might represent:

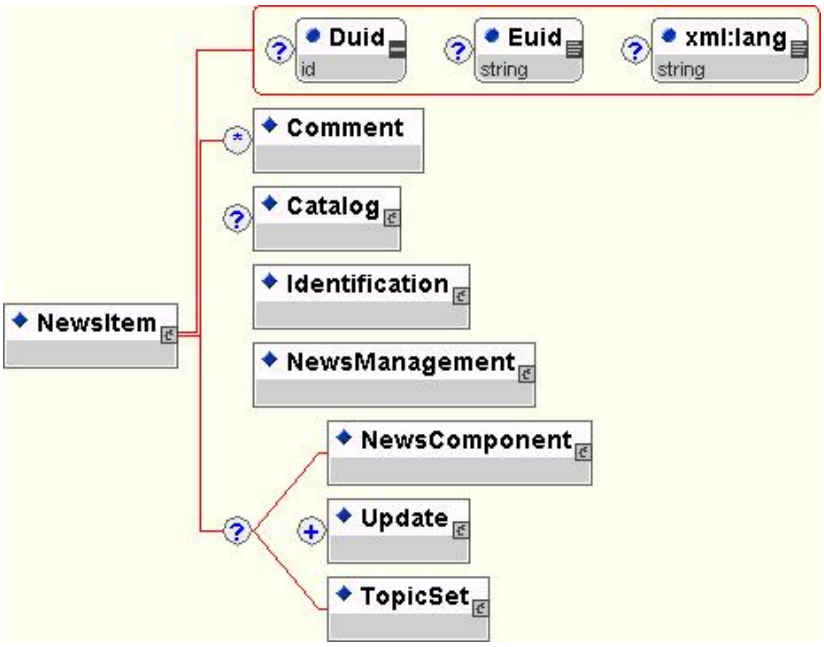
- A **unitary resource**, e.g. a text, a photo, a graphic, an audio or video clip.
- A **multimedia package** of different news objects, from mixed media, usually created by a multimedia desk. For example, in an editorial column composed of a text and some photos, a thumbnail picture can be included.
- A **collection** of related news objects; included content pieces usually share the same media (for example the five best pictures of the day). Such a collection may contain only links to other NewsItems, optionally with some labels or metadata; each pointer to a NewsItem is deemed to be replaced by the proper NewsItem element after resolution of the link.

5.2 Structure of a NewsItem

At a conceptual level, one can view a NewsItem as a NewsComponent with a name, a status and some other Identification and NewsManagement metadata.

A NewsItem can also be a collection of Update objects that represent some revision of large news objects in a compact way.

Controlled vocabularies - when defined as Topic Sets - also need identification and management features, and benefit from the NewsItem structure.



5.3 Formal identification of a NewsItem

It must be possible to positively identify a NewsItem as it moves through the news workflow, and is transferred from place to place and from system to system. NewsML therefore requires NewsItems to have a ***globally unique identifier*** in the form of a NewsIdentifier element.

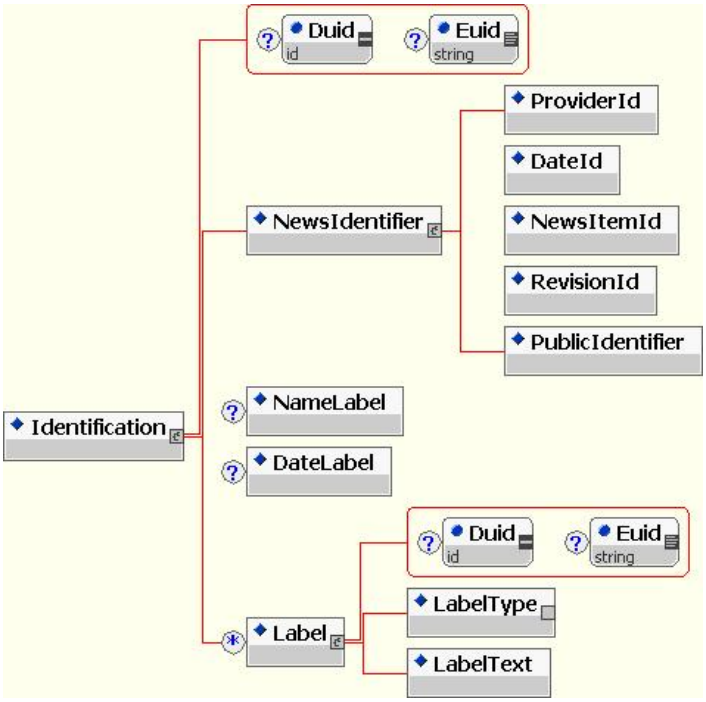
Features:

The identification element set offers the following features:

1. An identifier is unique among all potential information providers; all a provider must obtain to create a NewsItem is an internet domain name (see ProviderId); the IPTC maintains no central directory of providers.
2. An identifier includes a date (see DateId); the ownership of the ProviderId domain name at this date is required in order to avoid any possible duplication of identifiers.
3. A NewsML identifier can be created from real time or archive data, using a local system identifier or any token sensible for the provider, whether meaningful to humans or not; the only constraint on this token is its uniqueness in the provider's namespace for the given date (see NewsItemId).
4. A news item can be traced during its life cycle, using a simple revision mechanism (see RevisionId). A revision refers to an editorial decision to change the content of a NewsItem.
5. The URN (Uniform Resource Name) form of the identifier (see PublicIdentifier) is used for the creation of reliable links between NewsItems (see Links between news items), in a location independent manner.
6. Human understandable identifiers can be added to the news item, for easy cataloguing (see NameLabel, DateLabel and Label).

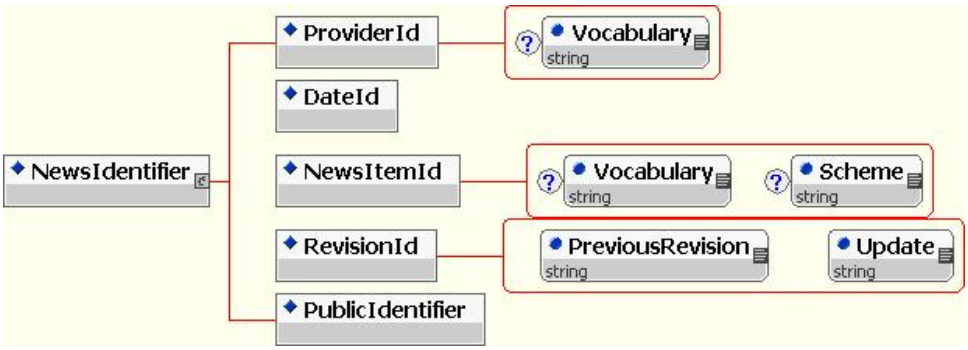
5.3.1 Identification

The **Identification** structure contains a set of elements:



5.3.2 NewsIdentifier

The NewsIdentifier has four component sub-elements – ProviderId, DateId, NewsItemId and RevisionId – and a PublicIdentifier, which concatenates all four components in a single string. The NewsIdentifier provides a globally unique identifier for a NewsItem. Providers must therefore ensure that no two NewsItems carry the same ProviderId, DateId, NewsItemId and RevisionId. If a NewsItem is re-created after a change in content, however slight, a new RevisionId should be allocated to the new version.



A quick description of the four **NewsIdentifier** elements:

Element	Description
ProviderId	The provider identifier, usually in the form of an internet domain name (e.g. “iptc.org”).
DateId	A date identifier, without mention of time, in ISO 8601 basic format (CCYYMMDD, ex. “20031231”).
NewsItemId	A news item identifier, unique in the provider’s namespace for a given date.
RevisionId	A positive integer. The default value is ‘1’, and its value is incremented when revisions occur.

5.3.2.1 ProviderId

The use of an Internet domain name is strongly recommended. The organisation that provides the NewsIdentifier must own the Internet domain name that is used as ProviderId at any date that appears in the DateId (see below). This will ensure that the identity of the provider can be inferred unambiguously from the full NewsIdentifier.

> **Going deeper:**
> **ProviderId** may be different from an internet domain name, and may be drawn from a controlled vocabulary via the Vocabulary attribute; see the NewsML Functional Specifications for more information.

5.3.2.2 DateId

The combination of ProviderId and DateId serves to uniquely unambiguously identify the organisation that is allocating the URN.

Because the DateId is part of the formal identification of the NewsItem, it must remain the same through successive revisions of the same NewsItem. In practice, the date identifier is often the date of creation of the NewsItem.

> **Note on ProviderId and DateId as part of a NewsItemId:**
> NewsItemIds are intended to be unambiguous for all time. The preferred value of ProviderId field is an Internet domain name that is owned by the provider of the NewsItem - but the owner of a domain name (the registrant) can change over time. The initial intention of the DateId portion of the NewsItemId was to disambiguate the ProviderId by specifying a point in time at which the Provider in question owned the URL. Thus, even if two Providers owned the same URL, they would own it at different times and hence use different values in the DateId field.
>
> After the publication of NewsML 1.0, it was noticed that there was a difference in wording between the NewsML Functional Specification and RFC 3085 describing the structure of a NewsML Public Identifier (URN). In conformance with some interpretations of the Functional Specification it had become current practice for certain providers to use the original creation date of the NewsItem as DateId. This contradicted the original wording of the RFC 3085, which stated that for a given provider the combination of ProviderId and DateId should be unique. The IPTC subsequently decided to recognise past and present practice of its members by changing the wording of the RFC so that the combination of ProviderId and DateId is described as being unambiguous, rather than unique, and hence not only can providers choose any date on which they own the domain name in question to populate the DateId field, they can also use different dates in different NewsItems. Different versions of the same NewsItem must, however, contain the same DateId.
>
> As a consequence, two NewsItems with the same ProviderId and DateId can be stated with confidence as being from the same provider. Two NewsItems with the same ProviderId but different DateIds may or may not be from the same

provider. Whether they are from the same provider cannot be determined by syntactic analysis of the NewsItemId. In practice domain names of major providers do not change hands frequently. Processors of NewsML that need to determine whether two news items are from the same provider may be able to use equality of the ProviderId together with other heuristics to determine this, when the DateId does not match. The Provider administrative metadata shall also be used for this purpose.

Implementers of NewsML should use one -- and only one -- value for DateId in all their NewsItems. But they may use other dates, such as the creation date of the NewsItem, providing they own the domain name used as ProviderId on that date.

5.3.2.3 NewsItemId

The NewsItemId is an identifier for the NewsItem. The NewsItemId must be unique among NewsItems that emanate from the same provider (i.e. with the same ProviderId and DateId). Within these constraints, the NewsItemId can take any form the provider wishes. It may take the form of a name for the NewsItem that will be meaningful to humans, but this is not a requirement.

Note about the syntax of NewsItemId:
As described below, the NewsIdentifier is used as part of a URN (PublicIdentifier), and also follows the rules of RFC 2141.
Thus, for example, the space character would appear as %20 and the "%" character itself would appear as %25.

Going deeper:
NewsItemId values may be constrained by a controlled vocabulary via the Vocabulary and Scheme attributes; see the NewsML Functional Specifications for more information.

5.3.2.4 RevisionId

The RevisionId is an integer indicating the specific revision of a given NewsItem. Any positive integer may be used, but if two instances of a NewsItem have the same ProviderId, DateId and NewsItemId, the one whose RevisionId has the larger value must be the more recent revision. A RevisionId of 0 is not permitted.

The **PreviousRevision** attribute must be present, and its value must be equal to the content of the RevisionId element of the NewsItem's previous revision, if there is one, and 0 if the NewsItem has no previous revision.

The **Update** attribute must be present, and its value must be one of:

Value	Description
U	The NewsItem contains an Update element or elements
A	The NewsItem consists only of a replacement set of NewsManagement data.
N	None of the above.

The use of the **Update** information is described in the **Chapter 6** "**The news management level – management strategies**".

5.3.3 PublicIdentifier

The **PublicIdentifier** of a NewsItem is a NewsML URN, a kind of URI (Uniform Resource Identifier) that identifies an object but doesn't locate it explicitly.

The existence of this URN enables the NewsItem to be referenced unambiguously by pointers from other XML elements or resources (e.g via the NewsItemRef element).

Within such pointers, if the RevisionId, its preceding ':' character and its following Update qualifier are omitted, then the pointer designates the most recent revision at the time it is resolved.

Note about RFC 3085:
A NewsML URN is obtained by the concatenation of the previously defined parameters, in the form defined by **RFC 3085**; the format of a standard NewsItem URN is:
"**urn:newsml:{ProviderId}:{DateId}:{NewsItemId}:{RevisionId Update}**".

where {x} means "the content of the x subelement of the NewsIdentifier".
The **Update** information is entered just after the RevisionId, e.g. '2U', '2A'.

RFC 3085, current official version of the "IETF Request For Comment" written by IPTC, is found at:

ftp://ftp.rfc-editor.org/in-notes/rfc3085.txt

This address is subject to change if a revision of this RFC is validated.

Note about the syntax of PublicIdentifier and its components:
PublicIdentifier is a URN and ProviderId, DateId, NewsItemId and RevisionId are used as part of it. Note that the set of characters that can be directly included within a URN is limited. The allowed characters are specified by the Internet Engineering Task Force (IETF) in its Request For Comments (RFC) number 2141. This document is available at

http://www.ietf.org/rfc/rfc2141.txt

Any character that is not within the permitted URN character set must be converted to a sequence of legal characters as described in RFC 2141.

Thus, for example, the space character in a URN would appear as %20 and the "%" character itself would appear as %25.

5.3.4 NameLabel, DateLabel and Label

The optional **NameLabel**, **DateLabel** and the generic **Label** are human readable strings used to help identify a news item.

Warning: No processing should rely on the content of these fields. No inference of uniqueness should be derived from them. They can be modified freely when the NewsItem is updated.

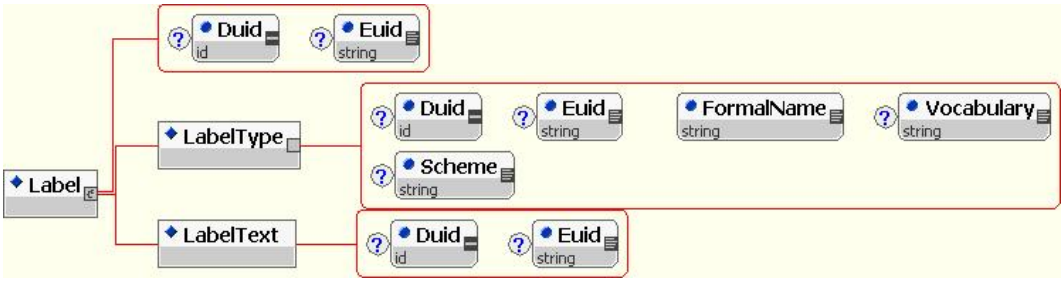
Limitation: Those labels have no multilingual representation.

The following are guidelines about the use of those labels:

NameLabel should be used as a shorthand reference to the NewsItem, given by its creator. An equivalent did exist in the IPTC IIM format, as Object Name (2:05).

DateLabel should express the date of publication of the news, as found in the *dateline* classical information. This date should be expressed in the main language of the NewsItem.

Label should be used when specific identifiers are given to NewsItems by a provider.



The **LabelText** sub-element holds the string label; the **LabelType** sub-element should belong to a proprietary controlled vocabulary, as the IPTC does not maintain such a TopicSet (see NewsML Metadata / Controlled Vocabularies). For example, a code representing the location of original transmission can be given to a picture by a photographer in the field before the picture is globally identified in the provider's namespace as a NewsItem. This name will later help the photographer to check if the picture has been correctly transmitted; an equivalent did exist in the IPTC IIM format, as Original Transmission Reference (2:103).

Example 5.3.4:

```
<NewsItem>
    <Identification>
        <NewsIdentifier>
            <ProviderId>iptc.org</ProviderId>
            <DateId>20030602</DateId>
            <NewsItemId>25ab03</NewsItemId>
            <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
            <PublicIdentifier>urn:newsml:afp.com:20020602:25ab03:1</PublicIdentifier>
        </NewsIdentifier>
        <NameLabel>IPTC 2003 AGM Aarhus</NameLabel>
        <DateLabel>6 june 2003</DateLabel>
        <Label>
            <LabelType FormalName="CodeName" />
            <LabelText>AAR01</LabelText>
        </Label>
    </Identification>
    ...
</NewsItem>
```

5.4 NewsItem and storage

If a system receives a NewsML envelope with several NewsItems in it (directly in the NewsML envelope or embedded in other NewsItems), it is recommended to "flatten" this structure and store independently as many NewsItems as transported in the NewsML envelope; each NewsItem should contain only references to other NewsItems (NewsItemRef element), and no embedded NewsItems. This way, all NewsItems are managed (revisioned, updated) and processed independently.

There is some duality in NewsML with the aspects relative to storage, since a NewsItem may be stored in a database or in a flat file system.

5.4.1 Database storage

When stored in a database, the NewsItem in usually stored as a stand-alone identified object, without the NewsML envelope (which is just a transport envelope).

In such a case, it is recommended to use the NewsItem's PublicIdentifier (the URN) as primary access key for the object.

5.4.2 File storage

When stored as a file, a NewsItem is kept embedded in its NewsML envelope as a full NewsML instance.

In such a case, a NewsML container should be as simple a possible; that is: the NewsML element should contain only one NewsItem, along with the useful transport properties (perhaps the NewsService/NewsProduct).

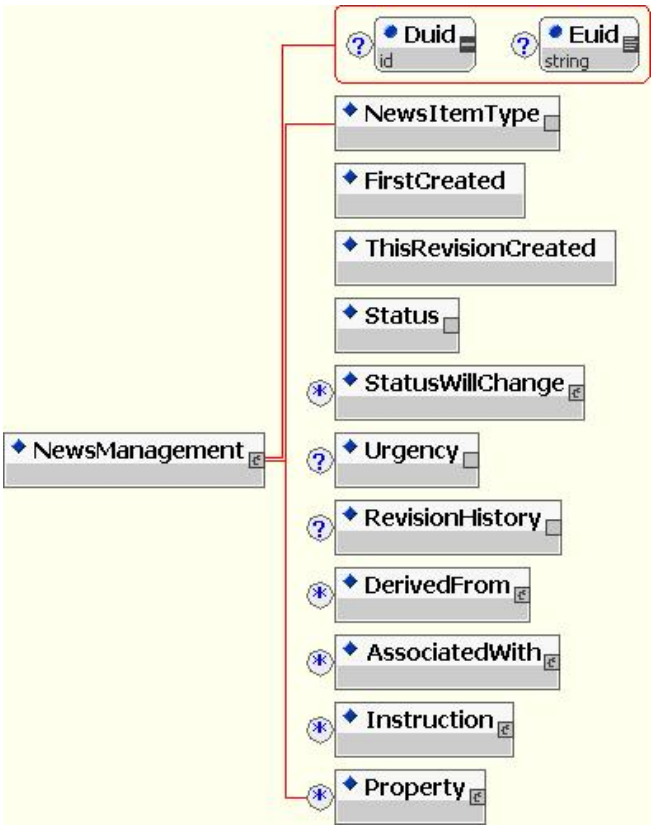
The file name given to the NewsML instance should correspond to the value of FileName (AdministrativeMetadata) found in the main NewsComponent of the NewsItem, if any.

> **Note:** The NewsItem could become a "first class citizen" for structured storage in the next major version of NewsML.

5.5 Management properties

The NewsManagement element provides information relevant to the management of a NewsItem: information about a NewsItem's type, history and status, as well as its relationship to other NewsItems, and any special instructions to be applied to it or additional properties it might have.

The **NewsManagement** structure contains a complete set of elements:



> **Going deeper:** detailed information about the **Property** element is given in the **Chapter 9**, "**Extension mechanisms**".

5.5.1 NewsItemType

The NewsItemType element contains an indication of the type of a NewsItem. The value of the FormalName attribute is a formal name for the news item type. The element is

empty; its value is represented via the FormalName and associated vocabulary attributes.

The IPTC maintains a set of NewsItemType values (e.g. '**News**', '**Data**', '**Advisory**', '**Alert**', '**Document**', etc.), along with their descriptions, in the "**topicset.iptc-newsitemtype.xml**" topicset.

> **Note about the Alert:**
> An alert requires the minimum of overhead so as to expedite transit and to allow the recipient to grasp the essential information as quickly as possible. A specific NewsItemType offers an efficient way of speeding this process. The message content may either be included in a NewsLine on as text in a ContentItem depending on the Provider's requirements.
>
> **Rule:** The use of the iptc-newsitemtype topic set is **normative** in the news agency community (i.e. news providers *must* use it and no other vocabulary).

5.5.2 FirstCreated

The date and, optionally, time at which a NewsItem was first created, expressed in ISO 8601 Basic Format.

5.5.3 ThisRevisionCreated

The date and, optionally, time at which the current revision of a NewsItem was created, expressed in ISO 8601 Basic Format.

5.5.4 Status

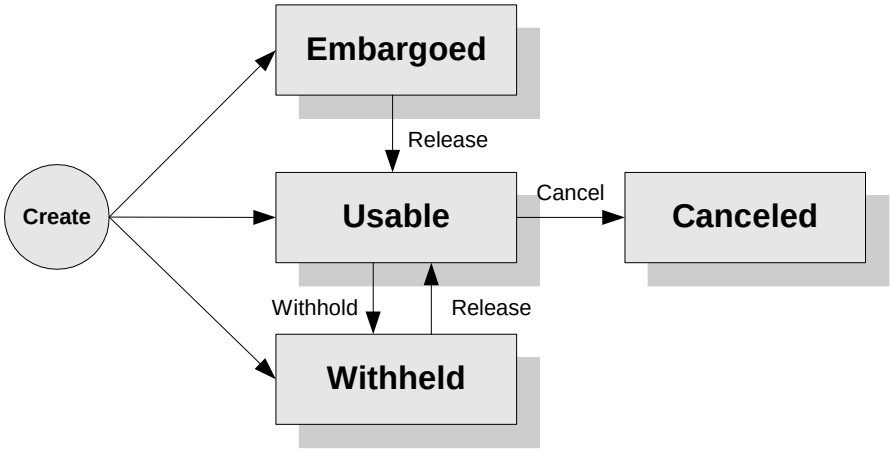
This element indicates the current status of a NewsItem. The element is empty; its value is represented via the FormalName and associated vocabulary attributes.

The IPTC maintains a set of Status values in the "**topicset.iptc-newsitemtype.xml**" topicset.

Value	Description
Usable	The NewsItem and its content may be published without restriction.
Withheld	Neither the NewsItem nor its content may be published until further notice.
Embargoed	Neither the NewsItem nor its content may be published until released for publication by the provider at a certain point in time.
Canceled	Neither the NewsItem nor its content may be used under any circumstances. If the NewsItem or its content has been published the publisher must take immediate action to withdraw or retract it, as may be legally necessary.

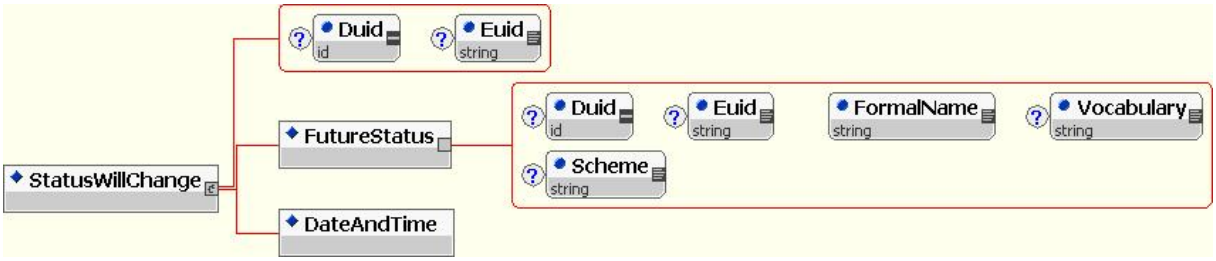
> **Rule:** The use of the iptc-status topicset is **normative** in the news agencies community (i.e. news providers *must* use it and no other vocabulary).

Their use is illustrated by the accompanying graphic. This depicts the state transition diagram reflecting the ways in which the Status values are intended to be used. Thus, upon creation of a NewsItem, allowed values of the Status element are "Usable", "Withheld" and "Embargoed". It is possible to withhold or cancel a "Usable" document; it is possible to release an "Embargoed" or 'Withheld" document. Once a NewsItem has had its Status set to "Canceled", it has reached a final state.



5.5.5 StatusWillChange

Advance notification of a status change that will automatically occur at the specified date and time. For example, an item with a Status of "Embargoed" might have a StatusWillChange element stating that the status will become "Usable" at a specified time. This is equivalent to announcing in advance the time at which the embargo will end and the item will be released.



Within StatusWillChange, the required **FutureStatus** element indicates the status the NewsItem will have at a specified future date. The element is empty; its value is represented via the FormalName and associated vocabulary attributes.

The required **DateAndTime** element indicates, using ISO 8601 Basic Format, the date or date and time at which the status will change.

As of version 1.1 multiple appearances of StatusWillChange are permitted. Multiple appearances of StatusWillChange allow for news management to provide a series of instructions about a NewsItem without the need for sending further NewsItems giving on status information. However, the DateAndTime values must follow a proper sequence of looking forward in time, and the series of Status update must follow the sate transition described above.

As examples, the following are valid:

Status	FutureStatus , DateAndTime(1)	FutureStatus, DateAndTime(2)
Usable	Canceled	-
Embargoed	Usable	Canceled

Where DateAndTime (2) comes after DateAndTime (1).

5.5.6 Urgency

An indication of the urgency of a NewsItem. The element is empty; its value is represented via the FormalName and associated vocabulary attributes.

5.5.7 RevisionHistory

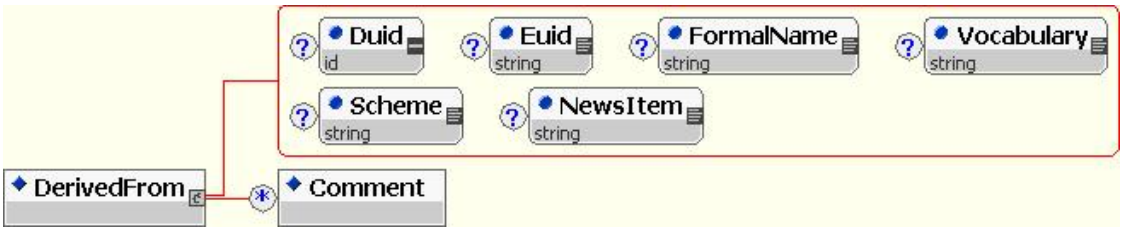
A pointer to a file containing the revision history of the NewsItem. The provider may choose the syntax and structure.

5.5.8 Derivedfrom

A reference to a NewsItem from which the current one is derived.

Added in NewsML v1.1, the type of derivation may be represented via the FormalName and associated vocabulary attributes.

The NewsItem attribute identifies the relevant NewsItem. Its value can be an http URL or a NewsML URN as described in the comment to PublicIdentifier.



The Comment (described in details in **Chapter 2**) can be used to provide informal additional information in natural language.

5.5.9 AssociatedWith

A reference to a NewsItem with which the current one is associated (for example, a series of articles, or collection of photos, of which it is a part).

Added in NewsML v1.1, the type of association may be represented via the FormalName and associated vocabulary attributes.

The NewsItem attribute identifies the relevant NewsItem. Its value can be an http URL or a NewsML URN.

Example 5.5.9:

```
<AssociatedWith FormalName="SeeAlso" NewsItem=" urn:newsml:reuters.com:20040302:ertgr25ab:1" />
```

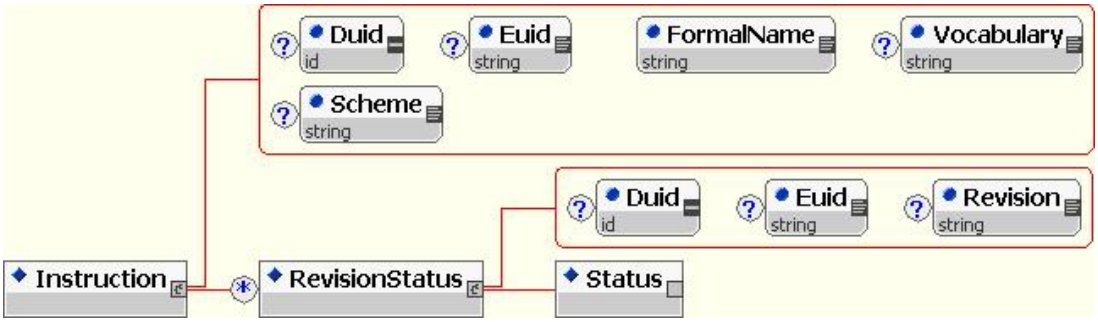
The Comment (described in details in **Chapter 2**) can be used to provide informal additional information in natural language, like a title associated with the target NewsItem (similar to the HTML alt-text) or a type/genre associated with the target NewsItem.

Example 5.5.9a:

```
<AssociatedWith FormalName="SeeAlso" NewsItem="urn:...">
  <Comment xml:lang="en" FormalName="Title">How NewsML makes you save money</Comment>
  <Comment xml:lang="en" FormalName="Genre">Analysis</Comment>
</AssociatedWith>
```

5.5.10 Instruction and RevisionStatus

A news provider may pass arbitrary instructions to a news subscriber via the Instruction element. The value of the **FormalName** attribute is a formal name for the Instruction, and its meaning is determined by a controlled vocabulary.that must be agreed upon by both parties in advance.

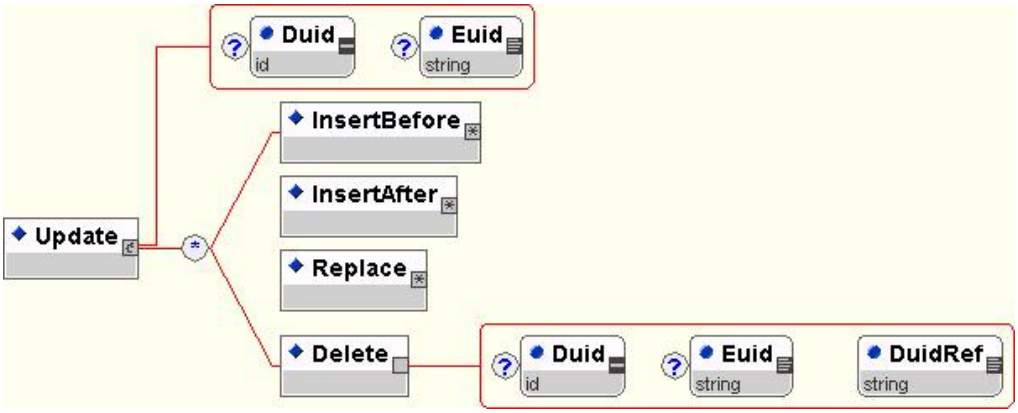


A special case of Instruction is an indication of the effect the current revision of a NewsItem has on the status of any previous revisions of the NewsItem that may still be on the recipient's system. In this case, it will contain one or more **RevisionStatus** elements.

A **RevisionStatus** element indicates the status of previous revisions as a result of the release of the current revision. The optional **Revision** attribute is a positive integer, equal to the RevisionId of the revision in question. If it is not present, then the status applies to *all* previous revisions, without exception.

5.6 Update elements

As part of the NewsItem element, the Update set of elements provides content relevant to the management of a NewsItem:



The Update element indicates a modification to an existing NewsItem. This can be an insertion, replacement or deletion. An Update element contains any number of sub-elements of the following kinds:

- **Delete**
- **Replace**
- **InsertBefore**
- **InsertAfter**

Each of those elements has a DuidRef attribute; its value should match the Duid of an element in the previous revision. This is the element to which the instruction applies.

The content model of those elements is ANY; that is, any well formed xml content may be included in it. In our scenario, this xml content must be a NewsML fragment. Providers must be careful that the result of the update of a NewsML document is a valid NewsML document.

The processing model of an update is described in **Chapter 6**, "**The news management level – management strategies**".

5.6.1 Update of Identification and NewsManagement properties

Modifications to these parts of the NewsItem can be made by issuing the NewsItem under the **current revision number**, with only the Identification and NewsManagement elements present. These will replace the previous Identification and NewsManagement elements in their totality.

Only the NameLabel, DateLabel and the generic Label element of the Identification set should be updated this way.

As seen above, when modified in such a way, a NewsItem is sent with the Identification/NewsIdentifier/RevisionId@Update attribute set to 'A'.

Note that the Update element cannot be used to modify the NewsManagement or Identification element, or any of their descendants. The use of the Update information is described in **Chapter 6**, "**The news management level – management strategies**".

Creators: S.Myles, L. Le Meur
Main contributors: J. Rabin, D.Gulija

NewsML 1.2 - Guidelines
Chapter 6: Management Strategies

6 The management level – management strategies

Often, news providers need to modify a news object that they have previously sent to a customer. For example, they may correct a headline, expand upon the body of a story or delete a piece of news altogether. This process of updating, deleting and modifying is known as "news management." Different news providers may have different news management policies. IPTC's NewsML standard provides sophisticated means for providers and their customers to implement a variety of procedures.

A news provider and its customers may implement news management in a number of different ways using NewsML. The most basic level is – essentially – *no* news management. In such a case a NewsItem is never modified after publication. A more sophisticated scenario involves the news subscriber maintaining an archive of published NewsItems. The news provider may issue new NewsItems, replace old NewsItems or delete old NewsItems in their entirety. The most sophisticated level of news management allows inserting, replacing and deleting parts of NewsItems – as well as the facilities provided at the "lower" levels of news management.

6.1 No Archive – Replacing one NewsML Document with another

In this scenario, the news provider does not use the status change, revision and update capabilities of NewsML.

6.1.1 Processing Requirements

The provider must still provide consistent identification of news items, but does not "version" NewsItems *per se*.

The news provider publishes new NewsItems as the information evolves, and may only "kill" news by sending a new NewsItem which describes this cancel instruction.

If the news provider publishes a collection of NewsItems (e.g. an index NewsItem with reference to the top ten NewsItems of the day), the management is made *ad-hoc*, e.g. using file names, de-indexing the NewsItems that are to be deleted, or by any other specific mean.

The news subscriber does not need to track the public identifiers for NewsItems – there is no need to try to match up subsequent NewsItems with previous NewsItems.

6.2 "Write Through" – Replacing and Deleting Complete NewsItems

In this scenario, the news subscriber maintains a news archive, i.e. a set of published NewsItems, and keeps track of the NewsItem identifiers. The news provider may issue subsequent NewsItems that replace or delete entries in the subscriber's archive. The subscriber must modify the news archive to reflect the changes specified by the provider.

6.2.1 Processing Requirements

The entire NewsItem is published, incorporating all changes that may have been made.

The news provider must track the ProviderId, DateId, NewsItemId and RevisionId associated with each NewsItem that it publishes. It does not need to track (or provide) element identifiers – Euids and Duids – in this scenario.

At a basic level, a provider may not publish any update of its NewsItems, and rely on the ProviderId/DateId/NewsItemId information to identify a NewsItem in a unique way; many news feeds are processed in such a way.

When publishing an update to a NewsItem, the provider must use a RevisionId that is a higher number before (usually by increments of one), and the PreviousRevision attribute is equal to the previous version's RevisionId. If the PreviousRevision is set to 0 then this is a new NewsItem to be added to the archive. The value of the Update attribute of the RevisionId element is set to 'N'. The other components of a NewsItem identity, ProviderId, DateId and NewsItemId must remain the same through successive revisions.

The news subscriber may replace the NewsItems in its archive, using the ProviderId/DateId/NewsItemId to identify which NewsItem to modify. If possible, a receiving system should keep the previous revision "hidden under" the new one, so that data mining of information updates is possible, and any explicit link to an old revision is still valid.

When deleting a NewsItem, the provider must use a NewsItem that will only contain the complete Identification and NewsManagement elements, and nothing else. The content of the RevisionId element is identical to that of the previous revision of the NewsItem, with the value of its Update attribute set to 'A'. The Status element indicates that the current status of the NewsItem is canceled (the IPTC provides a standard TopicSet for Status, which should be used by news agencies). The news subscriber must delete the entire NewsItem from its archive.

6.3 Updating, Deleting and Replacing Parts of NewsItems

In this scenario, the news subscriber maintains an archive of previously published NewsItems. The news provider may issue subsequent NewsItems that update, delete or replace constituents of NewsItems – at the element level – or that may replace NewsItems in their entirety. The subscriber must, therefore, track NewsItemIds for entire NewsItems and Duids and Euids for their constituent elements.

6.3.1 Processing Requirements

Upon receipt of a NewsItem from the news provider, the news subscriber must check the Update attribute of the RevisionId.

If Update is set to 'N 'and the PreviousRevision is 0, then this is a new NewsItem, which may be added to the archive.

If Update is set to 'N' and the PreviousRevision is greater than 0, then this NewsItem replaces the entire previous revision in the archive (see previous case).

If Update is set to 'A', then the subscriber must replace the Identification and NewsManagement data of the archived NewsItem (but the Update attribute value, which stays 'N' in the archived NewsItem). The content of the RevisionId element should be identical to the original one; the NewsItem should contain the complete Identification and NewsManagement elements, incorporating any changes, and nothing else.

If any other part of the NewsItem is modified in any way, the provider must use a RevisionId that is a higher number than before, and the PreviousRevision attribute should be equal to the previous version's RevisionId.

If Update is set to 'U', then the subscriber must process the Update element or elements contained in the NewsItem, updating the NewsItem in its archive.

In this case the NewsComponent subelement of the NewsItem is not included in the new document, but in its place, one or more Update elements are provided, indicating the modifications that have been made

It is the responsibility of the recipient to generate a new copy of the NewsItem on their system, by applying the Update instructions to the previous revision of the NewsItem, which they should already have, or be able to request from the provider. To generate the new revision of the NewsItem, each sub-element of each Update element is processed in turn, in the order in which they occur. The value of each subelement's DuidRef attribute should match the Duid of an element in the previous revision. This is the element to which the instruction applies.

In the case of **Delete**, the identified element is suppressed from the revised NewsItem.
In the case of **Replace**, the identified element is replaced by the content of the Replace element.
In the case of **InsertBefore**, the content of the InsertBefore element is added to the revision in front of the identified element.
In the case of **InsertAfter**, the content of the InsertAfter element is added to the revision after the identified element.

In distributed environments, special care should be taken when using differential updates, since there is no guarantee that a receiving system has successfully received or has archived the previous revision. Therefore, there has to be a way for a receiving system to request a full copy from the provider, something not possible in a broadcast environment.

In order to guarantee the robustness of an exchange mechanism, especially in text-based services, it is sometimes more convenient to simply send full NewsItems than differential updates.

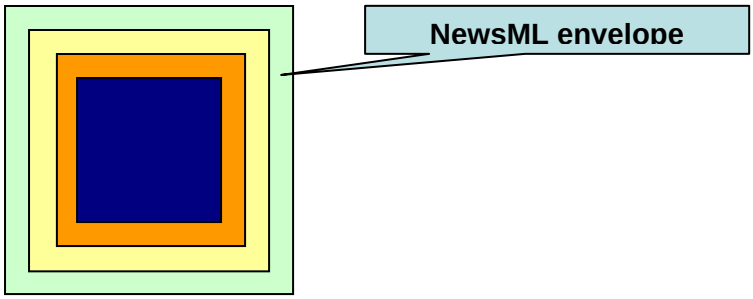
Creators: S.Myles, L. Le Meur
Main contributors: D.Gulija, J.Rabin

NewsML 1.2 - Guidelines
Chapter 7: The Exchange Level

7 The exchange level - NewsML envelope

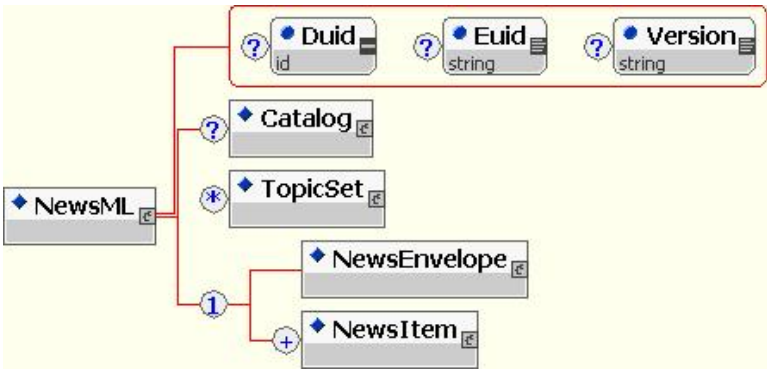
7.1 Context of the NewsML envelope

NewsML acts as a syndication format for news content represented as NewsItems. NewsItems are exchanged between editorial systems in NewsML envelopes. NewsML includes metadata that are useful for the interchange on news, but does not natively support workflow semantics; standard mechanisms may use the interchange metadata to allow routing of news-items through the editorial and production processes.



7.2 NewsML root element

The NewsML root element is a container for one or more **NewsItems**. A **NewsEnvelope** is used to provide information associated with the transmission of the NewsItems.



7.2.1 NewsML Version

The NewsML element supports a **Version** attribute, which is formatted as a decimal number (x.x).

The current version of NewsML is represented by the mandatory value '1.2'.

> **Note:** NewsML 1.0 didn't support the Version attribute; this feature appeared in NewsML1.1.

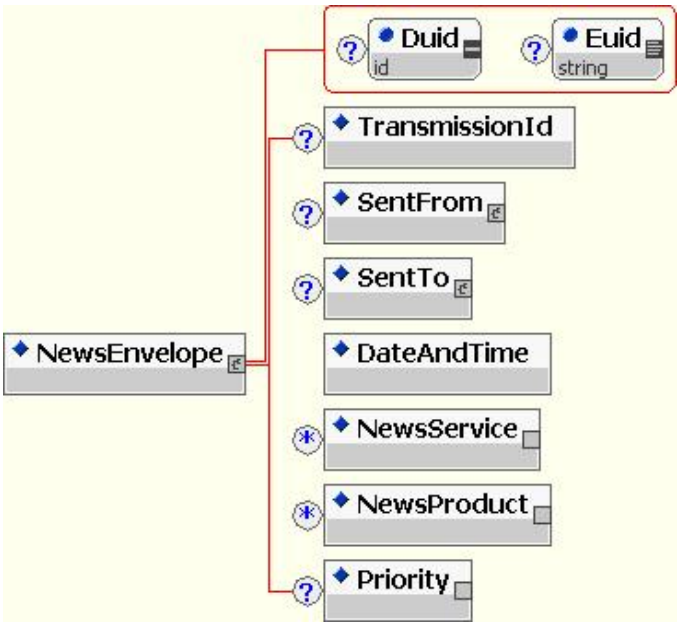
> **Going deeper:**

More information on the use of the NewsML/@Version attribute to control document compatibility with NewsML versions is given in the **Chapter 10**, **"Appendix"**.

7.3 NewsEnvelope

The NewsEnvelope element contains information about how the NewsML document is being used within a business workflow or contractual relationship between a news provider and receiver.

The **NewsEnvelope** structure contains the following set of elements:

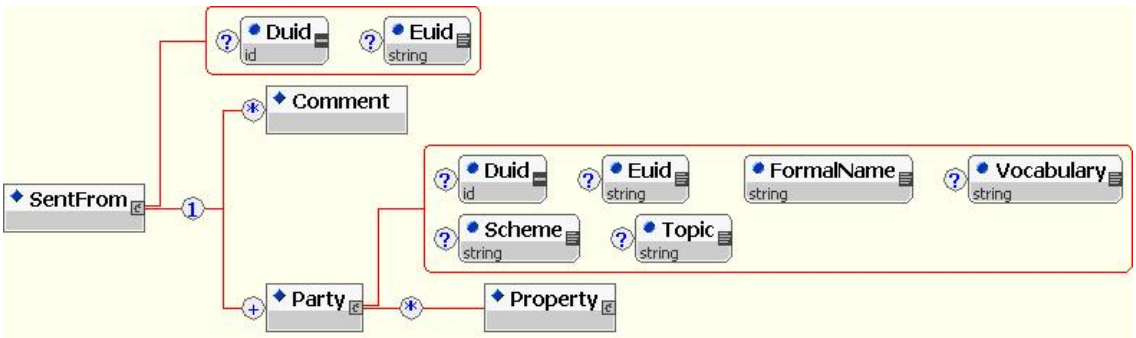


7.3.1 DateAndTime

The only mandatory element is **DateAndTime**, which represents the date and time of transmission of the NewsML instance in ISO 8601 basic format (e.g. '**20031231T000001+0100**').

7.3.2 SentFrom, SentTo

SentFrom is used to indicate the sender of the NewsML instance. The sender is identified in the Party sub-element through the use of the FormalName attribute, which indicates the party that is sending the instance.



SentTo is used to indicate the recipient (or set of recipients) of the NewsML instance, and gets the same structure as SentFrom. A recipient is identified in the Party sub-element. More than one Party may be expressed in SentTo, in order to represent a set of recipients.

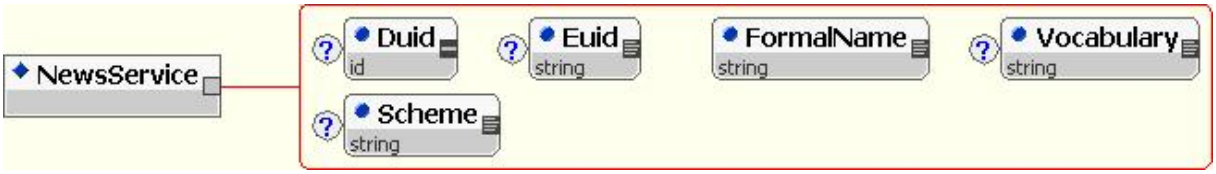
In a broadcast environment, SentTo may be used to represent logical groups of recipients, e.g. geographical areas like 'Europe', 'Italy', 'Roma&Florence'. The choice of syntax for such a "group address" is left to the news provider.

SentTo and SentFrom share the same model. The Party element is described in **Chapter 4, "Metadata about content – NewsComponent"**.

7.3.3 NewsService, NewsProduct

The NewsML functional specification doesn't precisely define the semantics of NewsService and NewsProduct. Definitions are found in the predecessor of NewsML, the IPTC Information Interchange Model (IIM).

The **NewsService** element allows a provider to identify the service that the attached NewsItems have been assigned to via the FormalName and associated Vocabulary attributes.



The **NewsProduct** element allows a provider to identify subsets of its overall service; this is used to provide receiving organisation data on which to select, route, or otherwise handle data.

NewsProduct and NewsService elements have the same attributes, which are used in analogous ways.

A NewsML envelope can be associated with several services and products. There will be usually one service only (the IIM didn't define repeatable service fields), and one or more products.

Example 7.3.3:
As an example a news agency could define a set of **NewsService** labels like "**TextWire**", "**PhotoWire**", "**Web**", and affect to **NewsProduct** its commercial products labels, like "**AsianTextWire**" or "**NouvellesDeFrance**".

7.3.4 Priority

The transmission **Priority** is useful when waiting queues are found in a transmission workflow. In such a queue, the NewsML envelopes of high priority should be put first.

The Priority is at a different level than the editorial importance given to a NewsItem, represented by the NewsManagement/**Urgency** element.

The IPTC maintains a set of Priority values (e.g. '**1**', '**2**', … '**8**', '1' standing for 'highest priority'), along with their descriptions in different languages. This topicset is published on the IPTC Web site.

Creator: L. Le Meur
Main contributors: Johann Lindgren, Jo Rabin

NewsML 1.2 - Guidelines
Chapter 8: Controlled Vocabularies

8. Controlled vocabularies for NewsML

NewsML provides a powerful metadata framework (see **Chapter 3, "The structure level — NewsComponents and metadata"**) where values of two basic types can be assigned to XML elements or attributes: controlled values and uncontrolled values.

A set of values controlled by any organisation is called a "controlled vocabulary". This chapter of the Guidelines explains the structure of IPTC's controlled vocabularies — known collectively as IPTC NewsCodes — and how they are implemented. In this document, IPTC NewsCodes are known as TopicSets.

A remark on the term *controlled vocabulary*: It means that the content of such a vocabulary — its set of terms — is controlled by an organisation; this could be a standardisation body like IPTC or a company acting as news provider. This term does not mean that a controlled vocabulary is controlled by anything in a NewsML instance; where IPTC wants to indicate that the value of a node (in terms of XML either an element or an attribute) must comply with a controlled vocabulary, we say that the node "is governed by a controlled vocabulary".

There are several good reasons for using controlled values for metadata:

- **To form well defined sets of normative values:**
  Some metadata must have one value out of a set of well defined values which are constant over time and can be used for automated processing, e.g. the value of the Status of a NewsItem (a NewsManagement element): statements like "can be used", "usable" and "use it" can be understood as equivalent by humans but not by processing software. Therefore a set of allowed values for Status must be defined as a controlled vocabulary and all NewsML instances must use one of these values.

- **To build a set of values common to many news providers and news consumers:**
  Some metadata are used to describe the content of NewsItems; they are primarily sub-elements to DescriptiveMetadata and ContentItem.
  In order to achieve powerful searches on NewsItems published by various news providers, metadata values must be shared at least by all services of a single news provider, and better if shared by many news providers. The most efficient way is to provide a method that is independent from the language used for the search terms; this can be done by the IPTC facilities. To that end, controlled vocabularies should be extensible over time and controlled by a news industry body: that is exactly what IPTC provides with its controlled vocabularies, e.g. standardized values for Genre, MediaType, Format or best known, SubjectCode.

- **To add value validation:**
  One of the key features of XML is its capacity to validate the structure of a specific instance by a reference to some schema definition language (DTD, XML schema, or any other type of schema) and signal any non-conformance.
  But the validation of the values associated with elements or attributes is completely impossible for a DTD-based validation and only possible to a certain extent by XML schema-based validation. Since NewsML 1.x relies only on a DTD — the XML Schema provided for the 1.x versions is "experimental" only — there is no way to validate values.

To provide this capability of value validation, controlled vocabularies and some specifications for their implementation were introduced in NewsML. This "controlled vocabulary mechanism" defines the elements or attributes which are governed by a vocabulary and the way to access the lists of allowed values associated with those elements or attributes.

What you need to know about controlled vocabularies:

- How to define what elements and attributes are governed — see section 8.1
- How to create a controlled vocabulary, and what it will do — see section 8.2
- How controlled vocabularies are built using topic sets — see section 8.3
- How to validate values — see section 8.4
- How to build a list of allowed values from a topic set based controlled vocabulary — see section 8.5

8.1 Defining elements and attributes governed by a controlled vocabulary

First, a basic distinction is to be made:

- The value of any FormalName attribute is governed **implicitly** — wherever this attribute appears, its value must be taken from a controlled vocabulary. (A definition of the FormalName attribute in the DTD: "A string of characters whose meaning is determined by a controlled vocabulary.")
- There are other elements and attributes which can be governed but this must be indicated **explicitly** — see more on how it is done below.

8.1.1 Elements governed implicitly

Currently — as of NewsML V 1.2 — these NewsML elements carry a FormalName attribute and are therefore governed implicitly:

AssociatedWith *)
Comment *)
Contribution
DerivedFrom *)
Format
FutureStatus
Genre
Instruction
LabelType
Language
MediaType
MetadataType
MimeType
NewsItemType
NewsLineType
NewsProduct

continued >>>

NewsService
Notation
OfInterestTo
Party
Priority
Property
Relevance
Role
Status
Subject
SubjectDetail
SubjectMatter
SubjectQualifier
TopicSet
TopicType
Urgency

For all these elements — except the ones earmarked with an asterisk (*) — the FormalName attribute is "required" by the NewsML DTD, making its use mandatory. For elements with the asterisk, the FormalName attribute is optional.

The meaning of the value of the FormalName attribute is different for different elements holding it and depends on the meaning of the element:

- For most of the elements the value of the FormalName attribute represents the *de facto* value for the element — these are the elements without an asterisk
- To some elements the FormalName attribute adds type information — what kind of content this element represents — and these are the elements with an asterisk.

As a developer of NewsML validation software, you can infer your algorithm from this rule:

- If a NewsML element has a FormalName attribute, the value of this attribute must be taken from a controlled vocabulary.
- To find the appropriate vocabulary, see section 8.3.

8.1.2 Elements governed explicitly

The NewsML way to specify an element or attribute as being governed by a controlled vocabulary is either:

- To define an XSLT-like pattern in the Context attribute of a DefaultVocabularyFor element inside a Resource element contained in a Catalog structure. If this pattern matches the element or attribute under examination by a NewsML validation software its value is governed by a controlled vocabulary and which one is indicated by the URN and URL elements of this Resource element.
- To set a value for the Vocabulary attribute of the specific element (currently, apart from the elements with a FormalName attribute, only two additional NewsML elements have a Vocabulary attribute: NewsItemId and ProviderId). This value indicates the controlled vocabulary.

As a developer of NewsML validation software you can infer your algorithm from this rule:

- If the current node matches the XSLT-like pattern found in the Context attribute of a DefaultVocabularyFor element — see details on this in section 8.4 - the governing controlled vocabulary is specified in the related Resource element (which is the ancestor of the DefaultVocabularyFor element).

[exclusive or]

- if the current element node has a Vocabulary attribute the governing controlled vocabulary is specified there.

More information on how the information about vocabularies is provided and how this information can be resolved to a list of values is explained in section 8.4.

8.2 "Controlled Vocabulary" Basics

The most essential definition of a controlled vocabulary is:
This is a list of values, each expressed as a string of characters.

But how is this "list of values" expressed and which notation convention has it to conform to? The most explicit information about this can be found currently in the comment on the FormalName attribute in the NewsML DTD: "A string of characters whose meaning is determined by a controlled vocabulary. The controlled vocabulary may (but is not required to) take the form of a NewsML TopicSet."

In fact IPTC currently has only outlined how to deal with TopicSets as controlled vocabularies, but the reader of these Guidelines is reminded that this is not mandatory. If a news provider develops another mechanism for providing the referential list of values for a controlled vocabulary it is up to him to implement this and to send his specifications to his customers.

IPTC's focus on TopicSet driven controlled vocabularies is the reason why these Guidelines explain only this type (in section 8.2.2).

8.2.1 Creation and maintenance of Controlled Vocabularies

8.2.1.1 Controlled Vocabularies provided by IPTC

Some of the controlled vocabularies provide metadata values that are important to the overall interpretation of a NewsItem. For consistency in handling NewsItems from multiple providers it is essential that common values are adopted for these key

parameters, namely **NewsItemType**, **Status**, **Priority**, **Location** and **Subject, SubjectMatter**, **SubjectDetail**, **SubjectQualifier, Location**.

Of the above controlled vocabularies those for **NewsItemType**, **Status, Location** and **Priority** are considered to be **normative** in the news agency syndication environment. Being normative means:

- A news provider using NewsML instances in their news services must use controlled vocabularies that hold *all* of the values from the IPTC vocabularies for these four properties.
- These controlled vocabularies can be extended, but this should be considered very carefully as this breaks the intention to have sets of values common to all news providers.

The IPTC has published controlled vocabularies (known as IPTC News Codes) taking the format of NewsML TopicSets for all of these, and it is recommended that providers adopt the published vocabularies as their defaults. Extensions may always be made but the extended TopicSets should be represented in the Catalog referenced for every NewsItem that employs any of the extended values. Alternatively, the extended values may be directly contained in every such NewsItem.

The IPTC has also published controlled vocabularies for various other elements but these vocabularies can be considered as templates or examples.

8.2.1.2 Controlled Vocabularies created by news providers

In order to ensure that users are able to route incoming NewsItems properly, providers who choose to make use of the optional child elements of NewsEnvelope must ensure that the following element attributes within NewsEnvelope are defined in controlled vocabularies provided to their users: **SentFrom**, **SentTo**, **NewsService** and **NewsProduct**.

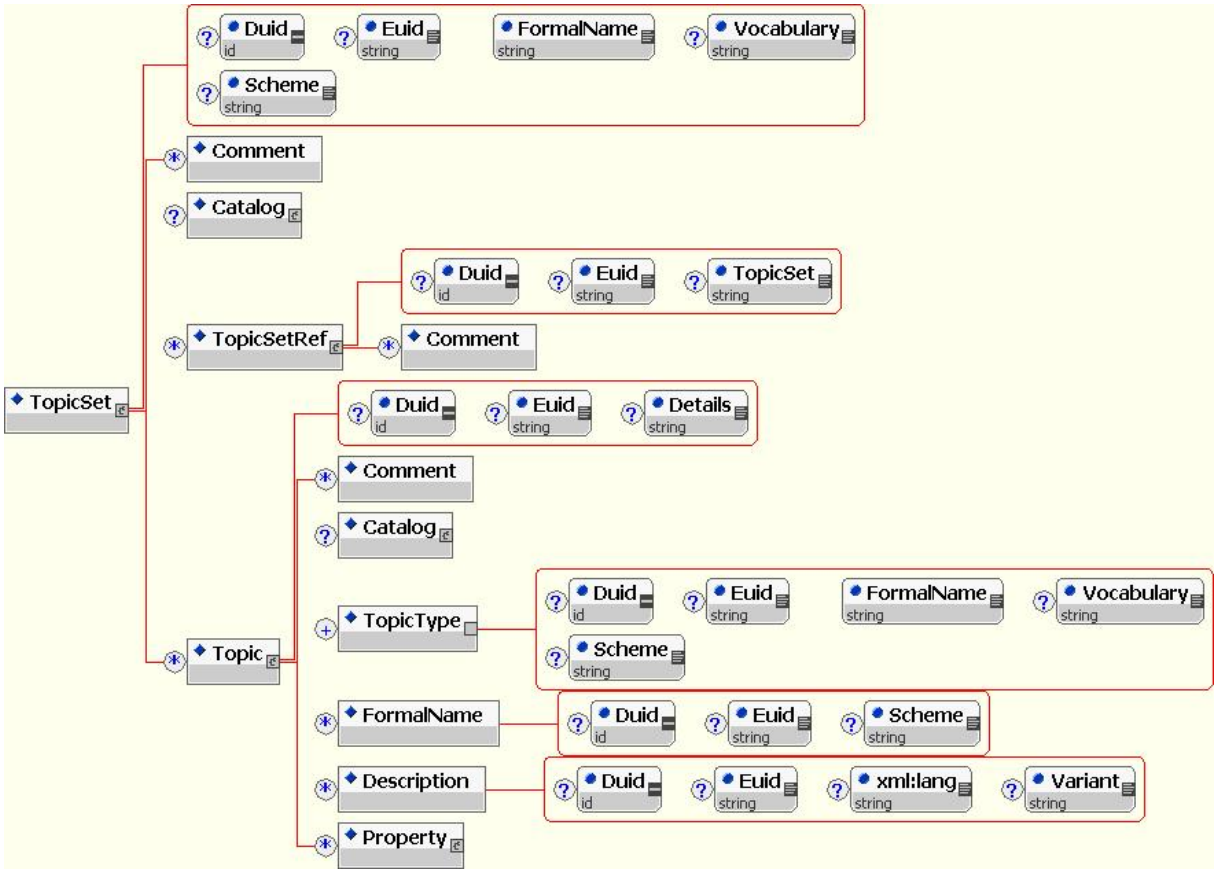
Other controlled vocabularies (e.g. **Role**) may be developed as necessary, to allow users to interpret aspects of the added value a provider makes to his news. The **OfInterestTo** controlled vocabularies is typically maintained by information providers. Publishers may add their own data to default controlled vocabularies as needed.

8.3 TopicSets as Controlled Vocabularies

A rather sober definition is:

A TopicSet is a set of named Topics.

8.3.1 TopicSet & Topic structure



8.3.2 What is a Topic?

From the NewsML specs, a Topic represents “any real-world thing or concept that can be referred to in a piece of news. Examples of a *topic* are the Iran-Iraq war, Tony Blair, Prime Minister of Pakistan, IBM, the United Nations, the Dyson vacuum cleaner, China, Kurdistan, Paris, the Kremlin, AIDS, aspirin, etc.”

> **Note**: The concept behind a Topic in NewsML is equivalent to the concept behind a topic as described in the “Topic Maps” specifications (find more on this at www.topicmaps.org).

A Topic element represents only a concept or a “real world thing” — its meaning could be explained in a Description element — but a Topic exposes nothing to the outside by which it can be referenced. The FormalName’s role is to be an externally available reference to a Topic. Such a formal name can be considered to be something like a name for a person, a trademark for goods or a term from common language for an abstract concept (i.e. Id in psychology). There could be several FormalNames assigned to a specific Topic, in this case each formal name must be drawn from a particular naming scheme and it must be unambiguous within this naming scheme.

Apart from formal names, the Topic element provides a type identifier, descriptions and properties.

8.3.3 How to specify a Topic

Take a look at the specification for the structure of a **Topic** element as shown in the figure in 8.3.1:

The most meaningful element to the use of a Topic is the **FormalName** element of a Topic (don't mix it up with the FormalName attribute of several other elements) as it represents the controlled term — the entry to the "list of values" — associated with the Topic: its content must be unique within its naming scheme, defined by the **Scheme** attribute of the element.

> **Note:** A NewsML validator software must signal an error if a FormalName value is used more than once within the scope of a specific naming scheme, i.e. two or more FormalName elements with the same Scheme attribute and the same content.

A short primer on this idea:

Concepts in the minds of people and objects that exist in the real world don't have a "name" — a verbal classifier — of their own from the beginning of time. Someone always assigns names to objects and concepts (i.e. nouns). This could either be an administrative entity like a national social security body that assigns Social Security IDs to human beings, or a municipal administration that assigns names or numbers to buildings or an editor of an encyclopaedia who defines that the concept of "material consisting essentially of protein, carbohydrate, and fat used in the body of an organism to sustain growth, repair, and vital processes and to furnish energy" [copyright Encyclopaedia Britannica] is called "food".

Real-world names relate to specific naming schemes; for example, the term "city hall" has specific but different meanings within the naming schemes of the City of New York and the Encyclopaedia Britannica.

On the other hand, it could easily be understood that there could be different formal names for a single topic: e.g.

```
<Topic>
    <FormalName Scheme="formal language use">father</FormalName>
    <FormalName Scheme="informal language use">daddy</FormalName>
</Topic>
```

There the concept of a person who is the male part of parents could be addressed by two totally different words.

A Scheme attribute is not mandatory for a Topic in NewsML. But if a FormalName element does not have a Scheme attribute, then its content is not defined within any specific naming scheme.

But remember that such a "schemeless" FormalName only matches against FormalNames without having a Scheme assigned also, hence a FormalName without a Scheme attribute must not be considered to be some kind of wildcard value that would apply to any requested naming scheme. Better to think of it as having a naming scheme without an identifier.

The use of the Scheme attribute is particularly useful in NewsML where several naming schemes are used in parallel for a single Topic: this is the case in the "ISO3166 country codes" topicset, where countries are defined with their 2 and 3 letter codes:

Example 8.3.3:

```
<Topic Duid="isoc392">
    <TopicType Scheme="IptcTopicType" FormalName="Country"/>
    <FormalName Scheme="ISO3166-alpha2">JP</FormalName>
    <FormalName Scheme="ISO3166-alpha3">JPN</FormalName>
    <Description xml:lang="en-GB" >JAPAN</Description>
</Topic>
```

Defining several FormalNames for a Topic should be done carefully, it should be considered that it must be the same object or concept that is addressed and don't mix this up with having the same name.
Example:

```
<Topic>
    <FormalName Scheme="biology">nut</FormalName>
    <FormalName Scheme="mechanics">nut</FormalName>
</Topic>
```

This Topic is completely correct for formal reasons, no validating software will show an error, but the equal formal names do not address the same concept as the Scheme attribute shows. The concept behind the first FormalName has that of a fruit of a tree, the second that of a mechanical part.

Each Topic has a mandatory **TopicType** element. Multiple TopicTypes may be defined for one Topic, but this feature is not used in IPTC TopicSets.
The FormalName attribute of the TopicType indicates the type of this Topic, therefore its value must be taken from an appropriate controlled vocabulary that defines a set of topic types.

Be aware that currently the value of the TopicType element is *not* used by the controlled-vocabulary-mechanism, or to put in it in other words: there is no way to select Topics from a TopicSet by their specific TopicType to form a controlled vocabulary for a specific use. So in fact the practical use of the TopicType is only for proper administration of Topics inside a TopicSet (grouping or selection).

Finally, the **Description** elements provide background information on the Topic. Editing content for Description elements one should have in mind that this applies to this topic for all FormalName elements, distinguished by different Scheme attributes.

Example 8.3.3a:
This SubjectQualifier Topic is an example of a full-blown IPTC-style description:

```
<Topic Duid="sr15000043">
    <TopicType Scheme="IptcTopicType" FormalName="SubjectQualifier"/>
    <FormalName Scheme="IptcSubjectCodes">15000043</FormalName>
    <Description xml:lang="en-GB" Variant="Name">sports facilities</Description>
    <Description xml:lang="en-GB" Variant="Explanation">Gymnasiums, stadiums or arenas where sports
events take place</Description>
    <Description xml:lang="en-GB" Variant="ChangeComment">added 2003-04</Description>
    <Description xml:lang="en-GB" Variant="ChangeVersion">9</Description>
</Topic>
```

The meaning of the various Description elements is determined by their Variant attributes:

- **Name** = verbal name for the topic in a language specified by the "xml:lang" attribute
- **Explanation** = descriptive information on the Topic, should not exceed 255 characters (for software compatibility reasons)
- **ChangeComment** = comment on why this Topic has been added or changed
- **ChangeVersion** = number showing for which TopicSet version this Topic was added or changed.

NOTE: the style described here for the Description elements is not required by any NewsML specification; it only illustrates the style chosen by IPTC for its own TopicSets created or updated in 2003 or later (TopicSets created earlier will be updated to this structure).

BUT this set of elements makes the use of TopicSets language independent: If the "Name" of a Topic is provided to the user and internally mapped back to the FormalName of the Topic this is a way to provide names in various languages, discriminated by

different “xml:lang” values, for the same FormalName, which does not allow for a distinction by “xml:lang”.

A proper “Explanation” of a term can avoid a misuse caused by a misunderstanding of their precise meaning. The explanation can be provided to the user as help or hint in the same language as the “Name” of the Topic.

8.3.4 How to build a TopicSet from Topics

A Topicset is a set of named Topics selected deliberately from the vast set of all imaginable topics to form a subset for a specific use. This specific use could be a specific operational purpose within NewsML — e.g. the Status TopicSet, which holds all valid values for the Status element — or to provide a well defined set of terms for descriptive metadata to classify content — e.g. the IPTC Subject Codes.

Hence a TopicSet can be considered to be a well-organised bag holding Topics; the bag is completely agnostic to its content. To emphasise this: a TopicSet does not add any meaning to Topics; it is simply a container for some of them — like a bottle that can be filled with water, wine or juice.

In NewsML the TopicSet element itself holds only sub-elements. Its FormalName attribute is currently of no relevance to the practical use of the TopicSet. It only represents an indicator of the type or nature of the Topics it contains.

> **Warning:** the value of the FormalName attribute of a TopicSet is implicitly governed by a controlled vocabulary, which is usually also a TopicSet. Hence the validation algorithm may enter a recursive loop. For this reason any implementation of a controlled-vocabulary mechanism based on NewsML TopicSets should have a “lock-out” for this recursion and should only check the value of the TopicSet’s FormalName attribute against the appropriate TopicSet *without* checking its FormalName against a controlled vocabulary again … and so on.

Therefore a TopicSet is built by simply adding Topic elements to a TopicSet element.

While building a TopicSet keep mind that the TopicType element is defined at the Topic level, and you must repeat the TopicType value for each included Topic even if all Topics of a TopicSet are of the same TopicType.

Different TopicTypes may exist in the same TopicSet, usually either for a Local Vocabulary (see 8.6), or as in the IPTC SubjectCodes TopicSet, where three TopicTypes (Subject, SubjectMatter and SubjectDetail) are assigned to different Topics.

The listing below shows an example of a full IPTC TopicSet, the one for the NewsManagement Status values:

Example 8.3.4:

```
<NewsML>
    <Catalog Href="http://www.iptc.org/IPTC/catalog/catalog.IptcMasterCatalog.xml"/>
    <NewsEnvelope> … </NewsEnvelope>
    <NewsItem>
        <Identification>…</Identification>
        <NewsManagement>…</NewsManagement>
        <TopicSet Duid="iptc.status" Scheme="IptcTopicType" FormalName="Status">
            <Comment xml:lang="en-GB">The current usability of a NewsItem.</Comment>
            <Comment>xml:lang attribute values updated</Comment>
            <Topic Duid="stat1">
                <TopicType Scheme="IptcTopicType" FormalName="Status"/>
                <FormalName Scheme="IptcStatus">Usable</FormalName>
                <Description variant=”Name” xml:lang="en-GB">Usable</Description>
                <Description variant=”Explanation” xml:lang="en-GB">The NewsItem and its content may be
published without restriction.</Description>
                <Description variant=”ChangeComment” xml:lang="en-GB">none</Description>
                <Description variant=”ChangeVersion”>0</Description>
            </Topic>
            <Topic Duid="stat2">
                <TopicType Scheme="IptcTopicType" FormalName="Status"/>
                <FormalName Scheme="IptcStatus">Embargoed</FormalName>
                <Description variant=”Name” xml:lang="en-GB"> Embargoed</Description>
                <Description variant=”Explanation” xml:lang="en-GB">Neither the NewsItem nor its content may be
published until released for publication by the provider.</Description>
                <Description variant=”ChangeComment” xml:lang="en-GB">none</Description>
                <Description variant=”ChangeVersion”>0</Description>
            </Topic>
            <Topic Duid="stat3">
                <TopicType Scheme="IptcTopicType" FormalName="Status"/>
                <FormalName Scheme="IptcStatus">Withheld</FormalName>
                <Description variant=”Name” xml:lang="en-GB"> Withheld</Description>
                <Description variant=”Explanation” xml:lang="en-GB">Neither the NewsItem nor its content may be
published until further notice.</Description>
                <Description variant=”ChangeComment” xml:lang="en-GB">none</Description>
                <Description variant=”ChangeVersion”>0</Description>
            </Topic>
            <Topic Duid="stat4">
                <TopicType Scheme="IptcTopicType" FormalName="Status"/>
                <FormalName Scheme="IptcStatus">Canceled</FormalName>
                <Description variant=”Name” xml:lang="en-GB">Cancelled</Description>
                <Description variant=”Explanation” xml:lang="en-GB">Neither the NewsItem nor its content may be
used under any circumstances. If the NewsItemor its content has been published the publisher must take immediate
action to withdraw or retract it, as may be legally necessary.</Description>
                <Description variant=”ChangeComment” xml:lang="en-GB">none</Description>
                <Description variant=”ChangeVersion”>0</Description>
            </Topic>
        </TopicSet>
    </NewsItem>
</NewsML>
```

Other details on the structure of TopicSets can be found in the IPTC paper “Detailed explanation of NewsML TopicSets” (by NSK, 2001), provided in the “Expert Zone” of the NewsML documentation.

8.3.5 Where to put a TopicSet element in a NewsML instance

A TopicSet element holding a controlled vocabulary may be located at different levels in NewsML by the structural definition in the DTD. These elements may contain a TopicSet element: NewsML, NewsItem and NewsComponent.

TopicSets are usually referred from NewsML instances, using a mechanism explained in depth in section 8.4. This mechanism accesses a TopicSet via the URN of the NewsItem in which the TopicSet is contained. For such a use, it is mandatory to embed the TopicSet as a direct child of the NewsItem element.

Example 8.3.5: (IPTC's SubjectQualifier TopicSet, cut to a single Topic)

```
<NewsML>
    <Catalog Href="http://www.iptc.org/IPTC/catalog/catalog.IptcMasterCatalog.xml"/>
    <NewsEnvelope>
        <DateAndTime>20030612T000000+0000</DateAndTime>
    </NewsEnvelope>
    <NewsItem>
        <Identification>
            <NewsIdentifier>
                <ProviderId>iptc.org</ProviderId>
                <DateId>20001006</DateId>
                <NewsItemId>topicset.iptc-subjectqualifier</NewsItemId>
                <RevisionId PreviousRevision="8" Update="N">9</RevisionId>
                <PublicIdentifier>urn:newsml:iptc.org:20001006:topicset.iptc-subjectqualifier:9</PublicIdentifier>
            </NewsIdentifier>
        </Identification>
        <NewsManagement>
            <NewsItemType Scheme="IptcNewsItemType" FormalName="TopicSet"/>
            <FirstCreated>20001030T120000+0000</FirstCreated>
            <ThisRevisionCreated>20030612T000000+0000</ThisRevisionCreated>
            <Status Scheme="IptcStatus" FormalName="Usable"/>
        </NewsManagement>
        <TopicSet Duid="iptc.subjectqualifier" Scheme="IptcTopicType" FormalName="SubjectQualifier">
            <Topic Duid="sr15000001">
                <TopicType Scheme="IptcTopicType" FormalName="SubjectQualifier"/>
                <FormalName Scheme="IptcSubjectQualifiers">15000001</FormalName>
                <Description xml:lang="en-GB" Variant="Name">men</Description>
                <Description xml:lang="en-GB" Variant="Explanation"/>
                <Description xml:lang="en-GB" Variant="ChangeVersion">0</Description>
            </Topic>
        </TopicSet>
    </NewsItem>
</NewsML>
```

But it is also possible to embed a TopicSet along with a NewsItem in the same NewsML instance. In this case the TopicSet is termed Local Vocabulary (see section 8.6) and the TopicSet element can be a child of a NewsML, a NewsItem or a NewsComponent element.

8.3.6 TopicSet extension using TopicSetRef

A TopicSet can be created as an extension of another predefined TopicSet, using the TopicSetRef element. TopicSetRef is a pointer to a TopicSet that is to be merged with the current one.

The TopicSetRef/@TopicSet attribute is a pointer to the relevant TopicSet. Its value can be an http URL, or a NewsML URN, or a fragment identifier consisting of a # character followed by the Duid of a TopicSet in the current document. The presence of a TopicSetRef child in a TopicSet has the effect that all the Topics in the referenced TopicSet are imported — by reference — into the current TopicSet.

When this merging would result in the existence of two Topics having the same formal name (i.e. same FormalName value and Scheme attribute), then those Topics are considered being identical and deemed to be merged to a single Topic. This merging of Topics need not be performed physically by the system, but processing these data must be done in exactly the same way as if the merging was physically performed. Merging two Topics results in creating a single Topic which contains all of the children of both, eliminating any duplicates.

8.3.7 Creation and Maintenance of Controlled Vocabularies / TopicSets

In the course of creating new controlled vocabularies, much consideration should be given to their content. The goal should be to define a comprehensive set of terms for a

specific purpose. But it is obvious that over time even the most carefully created controlled vocabulary requires some maintenance, usually the addition of new terms.

When the controlled vocabulary is defined as a TopicSet in a NewsItem, any revision of a TopicSet implies the update of this container NewsItem. The information in its NewsIdentifier element must be updated to reflect the new version — the RevisionId must be incremented — and the NewsML URN of the PublicIdentifier must be adapted accordingly. The date information in the NewsEnvelope and the NewsManagement elements must also be updated.

Finally, as the NewsML URN in the PublicIdentifier has been changed, the published Catalog file also needs an update — see section 8.4.1.7

Updated controlled vocabularies should be made backward compatible as much as possible. To achieve this, values should be added, but no values should be deleted. If this rule cannot be followed, old NewsML documents that use the previous version of the topicsets should not be required to pass vocabulary validity tests. To avoid failure, the non-compatible topicset file name should be changed, and the catalog file name should be changed. The new NewsML documents will refer to a new catalog file that refers to a new topicset file, and the old NewsML documents refer to an old catalog file that refers to an old topicset file (find more on Catalog issues in 8.4.1).

8.3.8 How to identify a Topic

As sections 8.3.1 and 8.3.3 show, a Topic can hold none, one, or several FormalName elements. This raises the question of how to determine if two different pairs of FormalName element/Scheme attribute reference the same Topic.

This can be done in a NewsML instance holding a TopicSet by detecting whether these two FormalName nodes are siblings under the same ancestor topic node or not — but this method is limited to the XML level of data representation.

If the TopicSet structure is transferred to a database, a locally unique identifier should be created for each Topic and be assigned to it. This ID can be used as a foreign key in a details table holding the FormalName/Schema pairs. Be advised that IPTC does not recommend a specific way to produce this unique identifier, and it is of a local scope for this database only.

8.4 Validating values — Controlled Vocabularies in action

The key issue with validating values is how to apply a controlled vocabulary to the value of a governed NewsML node to validate it.

Section 8.1 shows how to identify whether a specific NewsML node is governed at all.
Section 8.3 shows how to build a controlled vocabulary using the TopicSet format defined in NewsML to provide a controlled vocabulary for this node.
Finally, this section will show how those two types of information are brought to interaction.

It is essential to set up a relation between the NewsML instance representing a controlled vocabulary and an element or attribute that should be governed by this controlled vocabulary. There are two ways of doing this:

- defining the relation by a Resource entry of a NewsML Catalog — see section 8.4.1
- providing a direct pointer to the controlled vocabulary in the Vocabulary attribute of the element that must be validated — see section 8.4.2

8.4.1 Catalog

A Catalog in NewsML is the hub for integrating external controlled vocabularies into a NewsML instance. The Catalog provides a two-hop access for finding the appropriate controlled vocabulary for a specific node:

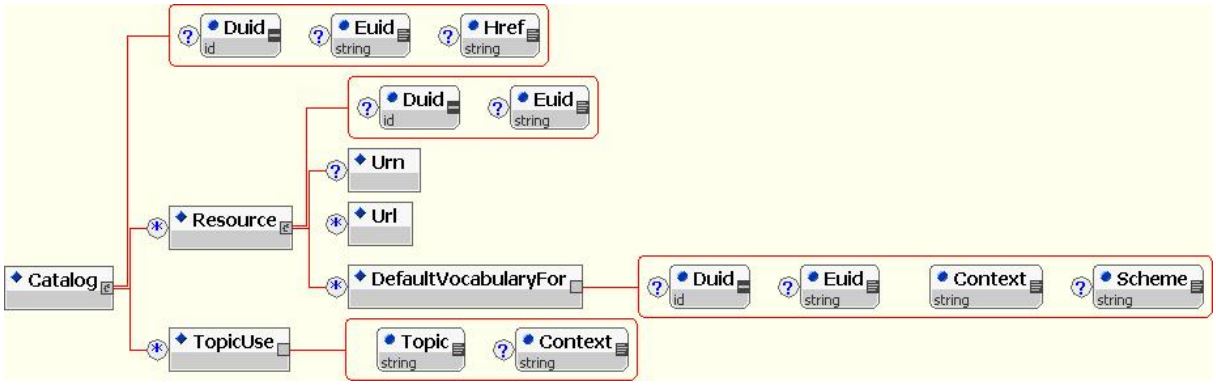
- First: Find and open the Catalog file.
- Second: Resolve the pointer to the resource found there, which represents the governing controlled vocabulary for this node. In the case of NewsML instance files holding TopicSets, find the physical location of the file to gain access to the TopicSets.

In a Catalog, a news provider must define the list of all controlled vocabularies he will use. This Catalog can be an adapted copy of the IPTC master Catalog or it can be a customized one.

The URL of the current IPTC master Catalog is:
http://www.iptc.org/IPTC/catalog/catalog.IptcMasterCatalog.xml

Important Note:
Never use the original IPTC master catalog file, since it points to TopicSet files on IPTC's web server (http://www.iptc.org/). As this server is not intended for production use of NewsML systems, access cannot guaranteed at all times. It is good practice to copy the TopicSet files from the IPTC server (http://www.iptc.org/IPTC/topicset/) to a local system and to adapt the Catalog file in a way that the Resource/URL values point to this local repository. See more details on the Resource element below.

8.4.1.1 Catalog structure



A catalog file is a specific NewsML file, with a set of Resource elements listed in a Catalog element, which is itself embedded in a DataContent element. Each Resource element of a Catalog represents a TopicSet, it shows:

- a **URN** element which shows the NewsML identifier (= PublicIdentifier) of the NewsItem holding the TopicSet
- a **URL** element which points to the NewsML file holding the TopicSet
- a **DefaultVocabularyFor** element with a **Context** attribute which holds the XSLT-like pattern nodes have to be matched against, and an optional **Scheme** attribute.
-

(Note: this applies only to controlled vocabularies defined as NewsML TopicSet files).

When a provider decides to extend or replace any IPTC defined TopicSet, he must define this in his own catalog. To do so, he can copy the IPTC master catalog, make the necessary modifications, and then save it to some server that can be accessed over the Internet. (See the "Important Note" in section 8.4.1)

In his specific Catalog, a provider can easily create a new Resource, based on the three parameters previously described.

If a provider customizes an existing IPTC topicset, he must exchange the original IPTC topicset reference in his Catalog for his own topicset reference. For better readability, provider specific topicsets should be moved to the top of the Catalog.

Warning: different Resource elements must always have different DefaultVocabularyFor/@Context values as in any other case a single node would have two or more controlled vocabularies assigned to — leaving the system confused.

Example 8.4.1.1: Catalog NewsML instance

```
<NewsML>
    <Catalog Href="http://www.iptc.org/site/NewsML/catalog/catalog.IptcMasterCatalog.xml"/>
    <NewsEnvelope>
        <DateAndTime>20010606T120000+0000</DateAndTime>
    </NewsEnvelope>
    <NewsItem>
        <Identification>
            <NewsIdentifier>
                <ProviderId>provider.com</ProviderId>
                <DateId>20010606</DateId>
                <NewsItemId>ProviderCatalog</NewsItemId>
                <RevisionId PreviousRevision="2" Update="N">3</RevisionId>
                <PublicIdentifier>urn:newsml:providr.com:20010606:ProviderCatalog:3</PublicIdentifier>
            </NewsIdentifier>
        </Identification>
        <NewsManagement>
            <NewsItemType FormalName="Catalog"/>
            <FirstCreated>20010606T120000+0000</FirstCreated>
            <ThisRevisionCreated>20030701T120000+0000</ThisRevisionCreated>
            <Status FormalName="Usable"/>
        </NewsManagement>
        <NewsComponent>
            <ContentItem>
                <DataContent>
                    <Catalog>
                        <!--Within the scope of an invocation of this Catalog, the following Provider
vocabularies will serve as default vocabularies-->
                        <!--Start of Provider vocabularies-->
                        <Resource>
                            <Urn>urn:newsml:iptc.org:20020707:topicset.provider-format:3</Urn>
                            <Url>http://www.provider.com/dtd/topicsets/topicset.provider-format.xml</Url>
                            <DefaultVocabularyFor Scheme="IptcFormat" Context="Format"/>
                        </Resource>
                        ------------------------------------------------------
                        <!--Within the scope of an invocation of this Catalog, the following IPTC vocabularies
will serve as default vocabularies-->
                        <!-- for specific NewsML elements and attributes as declared in the Context attributes
of the DefaultVocabularyFor elements-->
                        <!--Start of IPTC vocabularies-->
                        <Resource>
                            <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-confidence:1</Urn>
                            <Url>http://www.iptc.org/site/NewsML/topicsets/topicset.iptc-confidence.xml</Url>
                            <DefaultVocabularyFor Scheme="IptcConfidence" Context="@Confidence"/>
                        </Resource>
                        <Resource>
                            <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-genre:1</Urn>
                            <Url>http://www.iptc.org/site/NewsML/topicsets/topicset.iptc-genre.xml</Url>
                            <DefaultVocabularyFor Scheme="IptcGenre" Context="Genre"/>
                        </Resource>
                        ------------------------------------------------------
                    </Catalog>
                </DataContent>
            </ContentItem>
        </NewsComponent>
    </NewsItem>
</NewsML>
```

8.4.1.2 Reference to a Catalog in a NewsML instance

The *Catalog* used to validate element and attribute values of a specific NewsML instance must be referenced in this NewsML instance.

This can be simply done at the top of all NewsML documents, using the NewsML/Catalog element with a URL reference to the remote provider specific catalog file.

Example 8.4.1.2:

```
<NewsML>
    <Catalog Href="http://www.provider.com/IPTC/catalog/ProviderCatalog.xml"/>
    ....
</NewsML>
```

8.4.1.3 Use of the Context attribute of Resource/DefaultVocabularyFor

The XSLT-like pattern of this attribute specifies which XML nodes of a NewsML instance will be governed by this specific controlled vocabulary.

The processing model for applying this match pattern to a node is identical to the one specified for XSLT v1.0 in chapter 5. Find more about the processing of a full NewsML instance in section 8.4.2.

Examples:

For an element with a mandatory FormalName attribute, the expression matches the FormalName attribute:
<DefaultVocabularyFor Scheme="IptcGenre" Context="Genre/@FormalName"/>
For a specific element:
<DefaultVocabularyFor Scheme="IptcProvider" Context="ProviderID"/>
For a specific attribute of any element:
<DefaultVocabularyFor Scheme="IptcConfidence" Context="@Confidence"/>

8.4.1.4 Scope of applying a Catalog's Context patterns in a NewsML instance

The Catalog element can be a child to various elements: NewsML, NewsItem, NewsComponent, ContentItem, TopicSet, Topic, AdministrativeMetadata, DescriptiveMetadata, RightsMetadata and Metadata.

The scope of an expression in the pattern of a Resource/DefaultVocabularyFor/@Context attribute is the *child* axis to the Catalog's ancestor element; it is not allowed to match any nodes outside this axis against this pattern.

A good practice (and the simplest way) is to define the Catalog as a child of the NewsML root element, as shown 8.4.1.2. As a result from this all nodes of this NewsML instance are within the scope of this Catalog.

But in some cases, such as the aggregation of NewsComponents created by different providers, it may be necessary to reference more than one Catalog in a NewsML package. In this case the Catalog will be defined at the topmost possible level of each subpart, usually a NewsItem or NewsComponent. Be aware that this could lead to confusion over what vocabulary should be used for a given element value. Elements in the tree always inherit Catalog information from the nearest Catalog node in the ancestor axis form the node-to-be-validated. As a result, the information in the "closest" Catalog overrides any from more distant Catalogs — see also 8.4.2.

8.4.1.5 Use of the Scheme attribute of Resource/DefaultVocabularyFor

The Scheme value in the DefaultVocabularyFor element defines which FormalName elements of a TopicSet's Topic are referenced: only those with the same Scheme attribute value are selected.

If the DefaultVocabularyFor element has no Scheme attribute, then the appropriate Topic must have a FormalName element without a Scheme attribute.

Note: All IPTC Topic/FormalName elements have a Scheme attribute, so the DefaultVocabularyFor/@Scheme attributes must be set to a proper value when used with an IPTC TopicSet.

8.4.1.6 Access to a Catalog — best practice

A Catalog and the TopicSet files defined and managed by a provider should be accessed by software validating a NewsML instance only from time to time, never for each validation cycle as this will degrade the performance of the server holding these files.

As the controlled vocabularies in the TopicSet files and the Catalog file referencing them are presumably updated not more frequently than daily, IPTC considers it good practice to download all these data to a local system not more than once a day, then build the list of values locally and validate against them.

An alternative method is shown in the next section.

8.4.1.7 Maintenance of a Catalog

When a provider releases an update of its controlled vocabularies (TopicSets), he could alert its users, who should download again this provider's TopicSets instantly to keep their NewsML validator software in sync with the provider's controlled vocabularies.

For proper version tracking, the NewsML URN revision identifier of the Catalog instance should be updated when a modification occurs in any included topicset.

8.4.2 Bringing validation by Controlled Vocabularies to action

The figure on the next page shows an example of the validation of a value assigned to a FormalName attribute. Each box reflects a file required for this action.

Example 8.4.2:

THIS IS THE NEWSML INSTANCE IN A NEWS FEED

```
<NewsML>
    <Catalog Href="http://www.iptc.org/IPTC/catalog/catalog.IptcMasterCatalog.xml"/>
    <NewsEnvelope>
        <DateAndTime>20000717T160000
            </DateAndTime>
    </NewsEnvelope>
    <NewsItem Duid="Crts">
        <NewsComponent Duid="Mlkc1">
            <DescriptiveMetadata>
                <SubjectCode>
                    <SubjectMatter FormalName="11007000"/>
                </SubjectCode>
            </DescriptiveMetadata>
… etc
```

THIS IS THE CATALOG FILE catalog.IptcMasterCatalog.xml

```
<NewsML>
    <NewsItem>
        <NewsComponent>
            <ContentItem>
                <DataContent>
                    <Catalog>
                        <Resource>
                            <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-subjectcode:11</Urn>
                            <Url>http://www.iptc.org/site/NewsML/topicsets/topicset.iptc-subjectcode.xml</Url>
                            <DefaultVocabularyFor Scheme="IptcSubjectCodes"
Context="Subject/@FormalName|SubjectMatter/@FormalName |SubjectDetail/@FormalName "/>
                        </Resource>
… etc
```

THIS IS THE NEWSML FILE topicsets/topicset.iptc-subjectcode.xml HOLDING THE TOPICSET ACTING AS CONTROLLED VOCABULARY

```
<NewsML>
    <Catalog Href="http://www.iptc.org/IPTC/catalog/catalog.IptcMasterCatalog.xml"/>
    <NewsEnvelope>
        <DateAndTime>20031126T120000+0000</DateAndTime>
    </NewsEnvelope>
    <NewsItem>
        <Identification>
            <NewsIdentifier>
                <ProviderId>iptc.org</ProviderId>
                <DateId>20001006</DateId>
                <NewsItemId>topicset.iptc-subjectcode</NewsItemId>
                <RevisionId PreviousRevision="10" Update="N">11</RevisionId>
                <PublicIdentifier>urn:newsml:iptc.org:20001006:topicset.iptc-subjectcode:11</PublicIdentifier>
            </NewsIdentifier>
        </Identification>
        <TopicSet Duid="IptcSubjectCodes.subject" Scheme="IptcTopicType" FormalName="SubjectCode">
…
            <Topic Duid="sr11007000">
                <TopicType Scheme="IptcTopicType" FormalName="SubjectMatter"/>
                <FormalName Scheme="IptcSubjectCodes">11007000</FormalName>
                <Description xml:lang="en-GB" Variant="Name">human rights</Description>
                <Description xml:lang="en-GB" Variant="Explanation">Rights entitled to be enjoyed by all
citizens universally</Description>
                <Description xml:lang="en-GB" Variant="ChangeComment">none</Description>
                <Description xml:lang="en-GB" Variant="ChangeVersion">0</Description>
            </Topic>
… etc
```

Activating controlled vocabularies step by step: (references to the example figure above are in dark blue)

- While traversing a NewsML instance, the parser should check for *each* element or attribute node to determine whether its value is governed by a vocabulary.
  See more on the background in section 8.1.
- "Usual suspects" are all FormalName attributes since they are governed by vocabularies implicitly. But the condition of the next step must be met anyway.
  If the algorithm shown in the next step fails to identify a controlled vocabulary for a FormalName attribute the validating system should signal an exception.
  (In this example the FormalName attribute of SubjectMatter element requires a controlled vocabulary)
- Check whether this node is governed by a vocabulary defined by a Resource entry of the nearest Catalog.
  Use this general algorithm for this procedure:
  - Proceed to the ancestor element of the current element.
  - If it has a Catalog element as its immediate child, see whether that Catalog contains a Resource element whose DefaultVocabularyFor child's Context attribute contains an XSLT-match-style pattern that is applied to the current node. More details on the processing model for pattern matching by XSLT can be found in the XSLT specification [http://www.w3.org/TR/xslt] in chapter 5. Be aware that if an expression in this pattern has a local scope this scope is relative to the node currently under investigation — the current node is the context node.
    If the pattern match succeeds the resource identified by that Resource element is the controlled vocabulary assigned to this node.
  - If the ancestor does not meet the above condition of providing a Catalog child, proceed to its ancestor and check the same condition.
  - Continue until a vocabulary is found, or the root element has been reached.
  - If this algorithm fails to identify a Resource element in a Catalog that serves as controlled vocabulary for this node this node's value is NOT governed by a controlled vocabulary.

  (In the example above this algorithm searches along the orange arrow and finally finds the Catalog element under the NewsML element.)
- When a Catalog's Resource element defines the TopicSet and the DefaultVocabularyFor/@Scheme defines the naming scheme, then no additional Vocabulary and Scheme attributes are needed for this node. In such a scenario, the Catalog is the definitive source of information.
  (In the example above there is a Resource with a pattern in its Context attribute matching SubjectMatter/@FormalName nodes — see blue arrow to the left pointing up — and a Scheme attribute of the same value as the one of the SubjectMatter element — see blue arrow to the right pointing down.)
- If there are Schema and Vocabulary elements in parallel with a valid controlled vocabulary Resource then check the values of these attributes by the rules outlined in section 8.4.2.1 below.
- When a controlled vocabulary Resource for this node has been identified definitely it must be resolved.
- Check whether the URN element shows a NewsML URN string. In this case the controlled vocabulary is a NewsML TopicSet, if not the controlled vocabulary is of an unknown type.
  (In the example above the URN string is a valid NewsML URN as defined by RFC 3085.)
- In case of a NewsML TopicSet:
  Access the TopicSet on the path given in the Resource's URL element and check first if the URN given in the Resource element and the URN of one the NewsItems holding a TopicSet are equal. In case they are not equal this is an error condition.
  (In the example above the resource pointed to by the URL comprises a NewsItem with the identical URN in its PublicIdentifier element as the URN in the Resource element of the Catalog file — see the red arrow pointing down.)

- Use the value of the currently parsed node and find a Topic in the TopicSet identified as controlled vocabulary where the FormalName element's content matches and the Scheme attribute of the FormalName element and the Scheme attribute of the DefaultVocabularyFor element of the Catalog file are equal.
  (In the example above there is one of the Topics with a FormalName element holding the same value of "11007000" as the FormalName attribute of the SubjectMatter element — and the Scheme values align too. See green and blue arrows.)
- If all this matches the value of the currently parsed node is valid.
  (The example above shows a reference to a valid Topic value.)

8.4.2.1 Use of the Scheme and Vocabulary attributes

An element with a FormalName attribute can **locally** define by which vocabulary (TopicSet) it should be governed. This is done via the Scheme and Vocabulary attributes.

If the element includes just a **Scheme** attribute, as in

<Role FormalName="Sidebar" Scheme="NewsPaper" />,

the implication is that one should find the closest appropriate Catalog to determine which TopicSet is the source of data for the FormalName attribute of the element — see 8.4.2 above. HOWEVER, the appropriate Topic MUST have a FormalName of "NewsPaper" out of the Scheme "NewsPaper". If such a Topic does not exist, there is an error. If the DefaultVocabularyFor declares a different Scheme, then the local Scheme declaration in the element OVERRIDES the one of the Catalog.

If the element includes a **Vocabulary** attribute, as in

<Role FormalName="Sidebar" Vocabulary="urn:newsml…" />,

then the implication is that this local declaration of an internal or external controlled vocabulary (TopicSet) OVERRIDES any Catalog entry.

The value of the Vocabulary attribute is an http URL or a NewsML URN, or the # character followed by the value of the Duid attribute of a TopicSet in the current document (Local Vocabulary).
To apply these variant types of the Vocabulary value one should have in mind:
- URL: it points to a NewsML instance containing only ONE NewsItem with ONE TopicSet element, so the reference is unambiguous.
- URN: the URN has first to be resolved to an URL by facilities provided by the news provider
- #: this is a local reference within the current NewsML instance.

When one uses the Vocabulary attribute, one must be explicit about the use of the Scheme attribute. In the above example, since a Scheme attribute was absent, the indication is that the Topic holding the FormalName element with #PCDATA set to "Sidebar" has NO scheme attribute.

If the Topic in the TopicSet referred to by the Vocabulary attribute does have a Scheme attribute, then it should be explicitly declared along with the Vocabulary attribute, as in:

<Role FormalName="Sidebar" Scheme="NewsPaper" Vocabulary=" urn:newsml…" />

Summary:
1. No Scheme and no Vocabulary attribute for an element: default to the values of the closest appropriate Catalog.

2. Only a Scheme attribute of an element: find the closest appropriate Catalog, but use the Scheme attribute value to potentially override any Scheme value retrieved from the Catalog.
3. Only a Vocabulary attribute of an element: this attribute indicates which TopicSet is to be used to search for appropriate Topics. Any FormalName of a Topic must not have a Scheme attribute assigned to be appropriate for validation since the source element does not have a Scheme attribute as well.
The presence of the Vocabulary attribute in an element overrides any Catalog declaration regarding that element.
4. Both a Vocabulary attribute and a Scheme attribute of an element: an explicit declaration of TopicSet and Scheme to find the appropriate Topic. This overrides any Catalog declaration.

Or one can use the following table as a remainder:

[C] use Catalog entry
[o] override Catalog entry

A source element with a FormalName attribute► Controlled Vocabulary TopicSet ▼	No Vocabulary attribute		With Vocabulary attribute	
	No Scheme attribute	With Scheme attribute	No Scheme attribute	With Scheme attribute
Scheme assigned to a FormalName element of a Topic	[C]	[o]	Error	[o]
No Scheme assigned …	[C]	Error	[o]	Error

Describing this by pseudo-code:

```
If ./@Vocabulary exists then
        (Do NOT search a Catalog/Resource/DefaultVocabularyFor value)
        appropriate TopicSet:
                ./@Vocabulary
        appropriate Topic:
        If ./@Scheme exists then
                match (./@FormalName = Topic/FormalName)
                and (./@Scheme = Topic/FormalName/@Scheme)
        Else
                match (./@FormalName = Topic/FormalName)
                and (if Topic/FormalName/@Scheme not exist = true)
        EndIf
Else
        (Search a Catalog/Resource/DefaultVocabularyFor value)
        appropriate TopicSet:
        The closest appropriate Catalog which has an XSLT-like pattern of
        DefaultVocabularyFor/@Context matching the current element.
        appropriate Topic:
        If ./@Scheme exist then
                matches to (./@FormalName = Topic/FormalName)
                and (./@Scheme = Topic/FormalName/@Scheme)
        Else
                If DefaultVocabularyFor/@Scheme exists then
                        match (./@FormalName = Topic/FormalName)
                        and (Topic/FormalName/@Scheme = DefaultVocabularyFor/@Scheme)
                Else
                        match (./@FormalName = Topic/FormalName)
                        and (if Topic/@Scheme not exist = true)
        EndIf
EndIf
```

8.4.3 When the Controlled Vocabulary is not a TopicSet

Controlled vocabularies can be based on the TopicSet mechanism. Currently, IPTC defines all its controlled vocabularies as TopicSets — with one exception: the Language element.

In the Language element, the list of values from a TopicSet is replaced by the rules of RFC 3066 (e.g. 'en-US', 'fr-BE'). The background for this step: There is a vast number of combinations of language codes and country codes, and independently maintaining this list would be impractical.

In this case the provider (or IPTC) must explicitly state what rule is to be followed: the Catalog still defines a Resource element for this vocabulary; the DefaultVocabularyFor selects the nodes that follow this rule, no NewsML URN is defined for this Resource indicating that the controlled vocabulary is of another type than an IPTC NewsML TopicSet.

An URL can additionally be used to reference an internet resource that "identifies" the external controlled vocabulary — which is not a TopicSet. This URL should point to a resource that provides information about the rules of this controlled vocabulary. In the case of the RFC 3006 the most logical identifier is the URL "http://www.ietf.org/rfc/rfc3066.txt".

This way of representing metadata values should be used with great care, as NewsML parsers might not be able to check the validity of those values.

8.5 Listing valid values of a Controlled Vocabulary

In the course of creating a NewsML instance all element and attribute values that are governed by a controlled vocabulary have to be populated with a value from the assigned controlled vocabulary.
This is a short outline how this could be achieved with TopicSets as controlled vocabularies:

- Specify — at design time or at run time by a decent selection mechanism of a NewsML system — which elements or attributes controlled vocabularies are assigned. A good practice is to use assignments by a catalog file. This Catalog holds Resource elements which point on one hand to the nodes that are governed by a specific controlled vocabulary by its DefaultVocabularyFor/@Context's pattern and on the other hand to the Resource where the NewsML TopicSet can be found, usually a URL.
- When it comes to the procedure of assigning a valid value to such a governed element or attribute, a list of valid values must be made available. To that end all FormalName element values of the Topics from the assigned TopicSet, acting as controlled vocabulary, must be retrieved and added to a list. A prerequisite for this is determining which naming scheme applies: either there is a Scheme attribute assigned to the DefaultVocabularyFor element of the Catalog's Resource element or to the node where this value should be applied to. In this case of "having a Scheme" only the FormalName elements having a Scheme attribute of the same value should be selected from the TopicSet. If there is no Scheme attribute assigned to the DefaultVocabularyFor element and the node where this value should be applied to then only the FormalNames *without* a Scheme attribute at all should be retrieved.
- Selecting a value from such a "list of values" should result into adding this value to the node in question.
  As a basic procedure, a reference to the Catalog file must be included in the NewsML instance.

8.6 Local (uncontrolled) vocabularies

Local Vocabularies are usually used when a provider wants a set of Topics (people, locations, events, organizations, objects…), logically associated with a NewsItem, to be attached physically to this NewsItem.

Note: Special consideration is required for using local vocabularies: As pointed out earlier in this chapter, the key advantage of a controlled vocabulary is that it is common to the news items of at least one news provider — and even better, from many news providers. Having this in mind, the notion of defining Topics locally is quite contradictory since any Topic declared at this level is only valid for the NewsComponent it is attached to. So if one wants to use the same Topic for several news items, one must repeat this local vocabulary for each.

The IPTC admits that this problem comes from the current inability of NewsML to allow for uncontrolled values for FormalName attributes. So if a specific term seems to be most appropriate for an element with a FormalName and this term is currently not in a controlled vocabulary, a news provider must add this term in a local vocabulary.

In local vocabularies, references to these topics are usually represented by descriptive metadata TopicOccurrence elements, and the Duid of the locally defined Topic is used as a TopicOccurrence/@Topic fragment identifier value.

Example:

```
<NewsComponent>
    <TopicSet FormalName="NewsTopics">
        <Topic Duid="topic1">
            <TopicType FormalName="Event"/>
            <FormalName>2002-FIFA-WC</FormalName>
            <Description xml:lang="en">2002 FIFA  World Cup - Soccer</Description>
            <Description xml:lang="fr">Coupe du Monde de Football 2002 - FIFA</Description>
            <Property FormalName="SubjectMatter" Value="15054000" />
        </Topic>
        <Topic Duid="topic2">
            <TopicType FormalName="People"/>
            <FormalName>Z.Zidane</FormalName>
            <Description xml:lang="en">Zinedine Zidane, Position: MID</Description>
            <Description xml:lang="fr">Zinedine Zidane, milieu de terrain</Description>
            <Property FormalName="SubjectMatter" Value="15054000" />
            <Property FormalName="Position" Value="MID" />
            <Property FormalName="DateOfBirth" Value="19720623" />
        </Topic>
    </TopicSet>
    ----------
    <DescriptiveMetadata>
        <TopicOccurrence Topic="#topic1" />
        <TopicOccurrence Topic="#topic2" />
    </DescriptiveMetadata>
</NewsComponent>
```

Remark: An alternative to this indirect integration of descriptive metadata is the use of Property elements in DescriptiveMetadata. It is quite easier to handle and is explained in **Chapter 9**, "**Extension mechanisms**".

Main Editor: Michael Steidl
Main Contributors: Takahiro Fujiwara, Darko Gulija, Laurent Le Meur, Jo Rabin

NewsML 1.2 - Guidelines
Chapter 9: Extension Mechanisms

9 Extension mechanisms

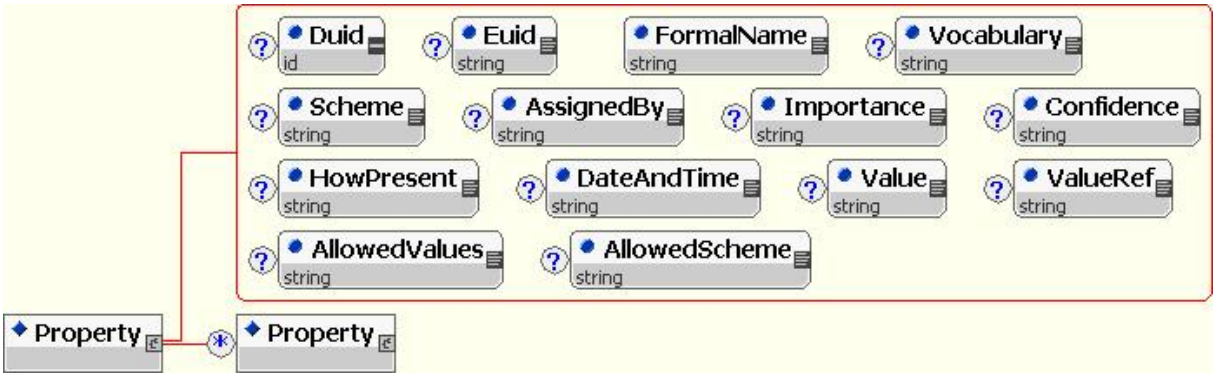
9.1 Structural Extensions

NewsML may be legitimately extended in the following ways:

- Default controlled vocabularies may be extended by adding additional entries inside existing Topic Sets, and new controlled vocabularies can be created. This is described in **Chapter 8**, "**Controlled vocabularies for NewsML**".
- Within the designated metadata sets (NewsManagement, AdministrativeMetadata, DescriptiveMetadata, RightsMetadata and Characteristics) additional metadata may be added using the Property element.
- New metatadata sets may be added by declaring a Metadata element associated with a MetaDataType and then adding the required number of Property elements.
- Within Party and Topic elements, additional metadata may be added using the Property element.

9.2 The Property element

The Property element is used to assert the value of some property on NewsManagement, AdministrativeMetadata, DescriptiveMetadata, RightsMetadata, Metadata, Characteristics, Party or Topic.



The property must be formally named and may contain sub-properties to handle complex properties.

An example of a simple Property is a representation of a country code:

```
<Property FormalName="Country" Value="DEU"/>
```

An example of a complex Property is a representation of a country, *identified* by its country code, and having its own (custom) properties:

```
<Property FormalName="Country" Value="DEU">
    <Property FormalName="Population" Value="80" />
    <Property FormalName="" Value="" />
</Property>
```

A complex Property may have no Value attribute:

```
<Property FormalName="Location" >
    <Property FormalName="Country" Value="US"/>
    <Property FormalName="CountryArea" Value="DC"/>
    <Property FormalName="City" Value="Washington"/>
    <Property FormalName="SubLocation" Value="The White House"/>
    <Property FormalName="WorldRegion" Value="North America"/>
</Property>
```

Note: a new Location descriptive metadata has been created in NewsMLv1.2 in order to replace the usage of the generic Property element as a location description.

9.2.1 FormalName, Vocabulary and Scheme attributes

The FormalName of a Property gives a type to the property. FormalName attribute has associated Scheme and Vocabulary attributes that follow the rules explained in **Chapter 8**, "**Controlled vocabularies for NewsML**".

The previous country Property may get such attributes that fully describe the vocabulary and scheme associated with the formal name:

```
<Property Vocabulary="urn:newsml:iptc.org:20001006:topicset.iptc-property:5" Scheme="IptcProperty"
FormalName="Country" Value="DEU"/>
```

9.2.2 Value, AllowedValues and AllowedScheme attributes

The Value attribute provides a string representation of the value of a Property. The Value attribute can be a free value, or have associated AllowedValues and AllowedScheme attributes, in the same way that FormalName has associated Scheme and Vocabulary attributes.

The AllowedValues attribute, if present, is a pointer to a controlled vocabulary that delimits the set of allowed values for the property. The pointer must reference either a TopicSet, or an internet resource that "identifies" an external controlled vocabulary (the vocabulary does not have to conform to the NewsML specification).

The NewsML DTD (v1.1) includes the AllowedScheme attribute in the Property element to designate the Scheme associated with the contents of the Value attribute of the Property. However, one can declare the TopicSet associated with the Value attribute via the AllowedValues attribute alone; in such a case the TopicSet in question does *not* have a Scheme attribute in the FormalName element which corresponds to the Value attribute.

For a match to be obtained within the controlled vocabulary both the AllowedScheme must match the Scheme in the vocabulary and the Value must match the FormalName in the vocabulary.

On the other hand, if AllowedValues is not used but the Catalog is relied on to indicate which TopicSet sources the values for the Value attribute of a Property, then a Scheme may be declared for the Topics providing the values of Value, via the DefaultVocabularyFor element or with the AllowedScheme attribute.

In other words, the Scheme attribute of DefaultVocabularyFor, when its Context attribute references a property value (Property/@Value), can replace the use of an AllowedScheme attribute in Property to indicate the use of a naming scheme for the value.

As an example:
Take a property with all attributes declared:

```
<Property
    Vocabulary="urn:newsml:iptc.org:20001006:topicset.iptc-property:5"
    Scheme="IptcProperty"
    FormalName="Country"
    AllowedValues="urn:newsml:iptc.org:20001006:topicset.iso-country:3"
    AllowedScheme="ISO3166-alpha3"
    Value="AND" />
```

The "short" variant

```
<Property FormalName="Country" Value="AND" />,
```

uses a **Catalog** that contains:

```
<Resource>
    <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-property:5</Urn>
    <Url>http://www.iptc.org/site/NewsML/topicsets/topicset.iptc-property.xml</Url>
    <DefaultVocabularyFor Scheme="IptcProperty" Context="Property"/>
</Resource>
...
<Resource>
    <Urn>urn:newsml:iptc.org:20001006:topicset.iso-country:3</Urn>
    <Url>http://www.iptc.org/site/NewsML/topicsets/topicset.iso-country.xml</Url>
    <DefaultVocabularyFor Scheme="ISO3166-alpha3"
    Context="Property[@FormalName="Country"]/@Value"/>
</Resource>
```

and a **Property TopicSet** that contains:

```
<Topic Duid="sprop5">
    <TopicType Scheme="IptcTopicType" FormalName="Property"/>
    <FormalName Scheme="IptcProperty">Country</FormalName>
    <Description xml:lang="en-GB">The country property found in the Location descriptive
    metadata.</Description>
</Topic>
```

and a **Country TopicSet** that contains:

```
<Topic Duid="isoc20">
    <TopicType Scheme="IptcTopicType" FormalName="Country"/>
    <FormalName Scheme="ISO3166-alpha2">AD</FormalName>
    <FormalName Scheme="ISO3166-alpha3">AND</FormalName>
    <Description xml:lang="en-GB" Variant="UPPER CASE">ANDORRA</Description>
</Topic>
```

9.2.3 ValueRef attribute

The ValueRef attribute gives a pointer to the value of the Property. This might be any piece of data referenced via a URI.

Example; use of a TopicMaps PSI:

```
<Property FormalName="Language" ValueRef="http://www.topicmaps.org/xtm/1.0/language.xtm#fr"/>
```

Note: The use of a NewsML TopicSet PublicIdentifier followed by a colon and the Duid of a Topic in the TopicSet is not recommended, especially as described in the Functional Specifications.

(e.g. do not use `urn:newsml:mydomain.com:20010101:Units:1#cm`)

One problem with this is that when the TopicSet is updated, its revision changes, and so its public identifier; if such a syntax is used, it must be without the inclusion of the optional revision identifier.

(e.g. `urn:newsml:mydomain.com:20010101:Units#cm`).

If both Value and ValueRef attributes are provided, then ValueRef identifies the actual value of the Property, with Value simply providing a string representation or mnemonic for it.

In this example, a label is given to the property, and more information if given in the referenced Topic:

```
<Property FormalName="Event" Value="Athens Olympic Games 2004" ValueRef="#topic1"/>
```

with a Local Vocabulary declaring:

```
<TopicSet FormalName="NewsTopics">
        <Topic Duid="topic1 ">
            <TopicType FormalName="Event"/>
            <FormalName>2004AthensOlympics </FormalName>
            <Description xml:lang="en-GB">Athens Olympic Games 2004</Description>
            <Property FormalName="BeginDate" Value="20040813" />
            <Property FormalName="EndDate" Value="20040829" />
        </Topic>
</TopicSet>
```

Another very interesting use of this ValueRef feature is the capability to reference any internet resource using a pure URL, as in:

```
<Property FormalName="Logo" ValueRef="http://www.mydomain.com/mylogo.jpg"/>
```

or:

```
<Party><Property FormalName="Biography" ValueRef="http://www.mydomain.com/mybio.xml"/></Party>
```

9.3 Use of Property element in Party and Topic elements

The Party element represents a person, company or organisation, and can be seen as a “complex type” in NewsML. The Party element is found in several elements: SentFrom & SentTo (NewsEnvelope), Source, Provider, Creator and Contributor (AdministrativeMetadata).

As explained in **Chapter 8, “Controlled vocabularies for NewsML”**, the Topic element represents any real-world thing or concept that can be referred to in a piece of news.

A Party element can point at a local or remote Topic via its @Topic attribute.

Topic and Party both support Property elements, which are metadata associated with the Party or Topic.

Example of Property elements associated with Topic elements:

```
<NewsComponent>
    <TopicSet FormalName="NewsTopics">
        <Topic Duid="topic1">
            <TopicType FormalName="Event"/>
            <FormalName>2002-FIFA-WC</FormalName>
            <Description xml:lang="en">2002 FIFA  World Cup - Soccer</Description>
```

```
            <Description xml:lang="fr">Coupe du Monde de Football 2002 - FIFA</Description>
            <Property FormalName="SubjectMatter" Value="15054000" />
        </Topic>
        <Topic Duid="topic2">
            <TopicType FormalName="People"/>
            <FormalName>Z.Zidane</FormalName>
            <Description xml:lang="en">Zinedine Zidane, Position: MID</Description>
            <Description xml:lang="fr">Zinedine Zidane, milieu de terrain</Description>
            <Property FormalName="SubjectMatter" Value="15054000" />
            <Property FormalName="Position" Value="MID" />
            <Property FormalName="DateOfBirth" Value="19720623" />
        </Topic>
    </TopicSet>
</NewsComponent>
```

Example of Property elements associated with a Party element:

```
<NewsComponent>
    <AdministrativeMetadata>
        <Creator>
            <Party FormalName="A.Legrand" >
                <Property FormalName="vCard.FN" Value="Arthur Legrand" />
                <Property FormalName="vCard.ORG" Value="FOO" />
                <Property FormalName="vCard.EMAIL" Value="amiller@foo.com" />
            </Party>
        </Creator>
    </AdministrativeMetadata>
</NewsComponent>
```

9.4 Extending metadata sets

9.4.1 Standard metadata sets

The AdministrativeMetadata, DescriptiveMetadata, RightsMetadata sets can be extended using Property elements.

As DescriptiveMetadata, using Property element is an alternative to the use of Local Vocabularies and TopicOccurrence elements. This solution only lacks the possibility to represent the description of the topic/property (as the Property element doesn't support any xml:lang attribute).

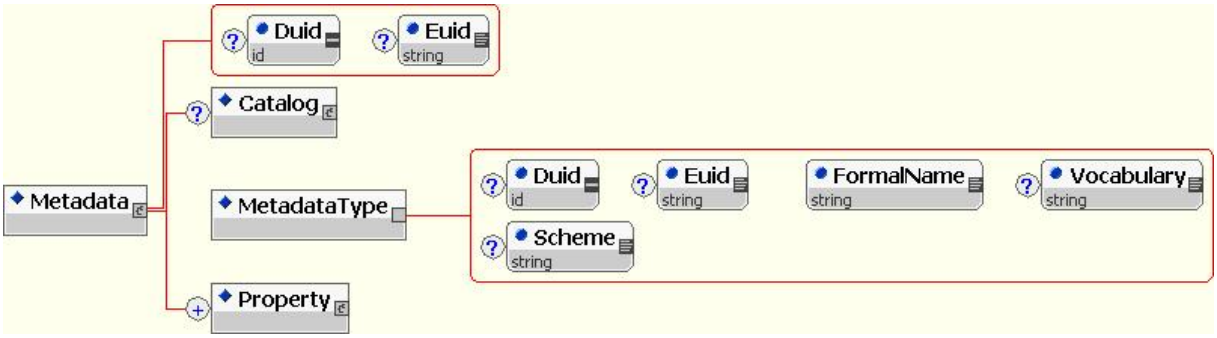
As an example, the sample shown in Chapter 8, "Controlled vocabularies for NewsML" to highlight the use of Local Vocabularies can be translated to:

```
<NewsComponent>
    <DescriptiveMetadata>
        <Property FormalName="Event" Value="2002-FIFA-WC">
            <Property FormalName="SubjectMatter" Value="15003000" />
        </Property>
        <Property FormalName="People" Value="Z.Zidane">
            <Property FormalName="SubjectMatter" Value="15003000" />
            <Property FormalName="Position" Value="MID" />
            <Property FormalName="DateOfBirth" Value="19720623" />
        </Property>
    </DescriptiveMetadata>
</NewsComponent>
```

9.4.2 Generic metadata sets

The generic Metadata set allows providers to define other kinds of metadata. This feature is useful when a provider wants to represent a set of metadata that doesn't fit in the standard categories (administrative, descriptive, rights); as an example one can represent a direct translation of the Dublin Core metadata set, or some specific publishing metadata, as freely defined Property elements.

The MetadataType element (via its FormalName, Vocabulary and Scheme attributes) gives a class to the metadata set.



Creator: L. Le Meur

NewsML 1.2 - Guidelines
Chapter 10 - Appendix

10 Appendix

10.1 XML encoding

Wherever possible, providers should use **UTF-8** for all document instances.

10.2 Format of dates

The date format chosen by the IPTC for NewsML is the **ISO 8601:2000** ***basic format*** (e.g. 20030705T160000+0100).

Note that this is not exactly the date format recommended by the W3C (e.g. 2003-07-05T16:00:00+01:00).

Truncated representations (the omission of higher order components where their presence is implied) are not currently allowed.

It is required to indicate the difference between local time and UTC, or to express the time of the day in Coordinated Universal Time (**UTC** i.e. GMT). When indicated, the representation of the difference is expressed in hours and minutes.

Examples:
20030705T150000Z stands for year 2003, fifth of July, three pm UTC time (using the Z [Zulu] designator as UTC flag; +0000 may be used instead).

20030705T160000+0100 stands for year 2003, fifth of July, four pm, western Europe (Paris) time.

For more information, check: http://www.pvv.org/~nsaa/8601v2000.pdf

10.3 The NewsML namespace URI

The definition of a namespace URI is needed in order to use **qualified names** (elements with a namespace prefix) for IPTC defined elements; this issue is important when specialized content is embedded in NewsML (at least to define NewsML as the default namespace). A namespace prefix is an abbreviation for a namespace URI.

The IPTC membership agreed to **change the namespace URI for each version**; this was perceived – along with the NewsML/@Version attribute discussed below - as a way to provide a strong version management of NewsML.

The NewsML namespace URI is currently specified as:

- **NewsML1.0 → urn:newsml:iptc.org:20001006:NewsML**
- **NewsML1.1 → urn:newsml:iptc.org:20021011:NewsML**
- **NewsML1.2 → urn:newsml:iptc.org:20031010:NewsML**

Note about the future syntax:
From NewsML v2, the NewsML namespace URI will be defined out of IPTC's URN namespace; ex: **urn:iptc:std:newsml:2.0:xmlns**

Example:

```
<NewsML xmlns="urn:newsml:iptc.org:20031010:NewsML">
    ...
    <NewsItem>
        ...
        <NewsComponent>
            ...
            <ContentItem>
                <DataContent>
                    <nitf xmlns="urn:newsml:iptc.org:20011012:NITF">
                        <body>
                            <body.content>
                                <p>Today, <person>Clinton</person> visited…</p>
                                <p><person>Al Gore</person> also attended the…</p>
                            </body.content>
                        </body>
                    </nitf>
                </DataContent>
            </ContentItem>
        </NewsComponent>
    </NewsItem>
</NewsML>
```

Other usages of the namespace URI are:

- Define a public identifier for the NewsML DTD;
- Identify the NewsML XSD schema.

Those usages are described in the next paragraph.

10.4 Validation of NewsML documents

10.4.1 Validation against the DTD

NewsML is a trademark of the IPTC and documents that are stated to be NewsML must be in conformance with one of the published DTDs (version 1.x).

Validation is switched on by declaring a DOCTYPE declaration which declares the NewsMLv1.2 system identifier (**urn:newsml:iptc.org:20031010:NewsML**) and the NewsMLv1.2 public identifier.

Of course, any reputable provider should ensure that his contract with his users stipulate that NewsML instances will be valid in accordance with an IPTC published DTD and they do not have to validate instances on receipt.

So the validation of NewsML instances is only useful for debugging purposes, and no DOCTYPE declaration is needed in the NewsML instances syndicated by a provider.

Note that even if NewsML instances contain a DOCTYPE declaration, the receiver should normally switch off validation (this is a configuration parameter of all XML parsers). Avoiding the validation of XML instances saves a great amount of processing power on the receiving side.

In the case a DOCTYPE declaration is used, the public identifier of a DOCTYPE declaration should not be a URL associated with the iptc.org Web site. Such an approach would create an unacceptable burden on the IPTC server and renders the provider and his users

at risk from any failure to access this server, as the IPTC server is not failsafe and does not form part of a guaranteed 24x7x365 service.

A safer approach is to remove the dependency on the IPTC server by using a local copy of the DTD either at the provider's site or one previously distributed to the users.

So the following DOCTYPE tags could be suitable ways of indicating the SYSTEM locations of a copy of the DTD:

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20031010:NewsML"
"http://www.provider.com/dtd/NewsML_1.2.dtd">
```

or

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20031010:NewsML"
"./dtd/NewsML_1.2.dtd">
```

Note: Using an abbreviated syntax without mention of a public identifier, as <!DOCTYPE NewsML> or <!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20031010:NewsML"> does end up with parsing problem on many systems, and thus should be avoided.

The proper validation of a NewsML instance which contains **specialized XML data** is obtained via the declaration of the embedded DTD as an entity reference or as a local DTD subset in the DOCTYPE declaration structure at the top of the document instance.

E.g. NITF content is declared via the following:

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20031010:NewsML"
"http://www.provider.com/dtd/NewsML_1.2.dtd" > [
<!ENTITY % nitf SYSTEM "http://www. provider.com/dtd/nitf-3-1.dtd">
%nitf;
]>
```

Note: This method has been already described in **Chapter 2**, "**The Content level – ContentItem**".

10.4.2 Validation against the XSD schema

The NewsML XSD schema is still experimental, and should not be used as a validation tool in a production environment.

In a schema environment, the namespace URI takes the place of the public identifier in the DTD world: it **identifies** the schema; each new XML standard is given such an URI.

The URI is defined in the XSD schema as a ***targetNamespace*** attribute, as in:

```
<xsd:schema targetNamespace=" urn:newsml:iptc.org:20031010:NewsML " …>
```

XML instances contain the namespace URI in the xsi:schemaLocation attribute; in a generic case, if 'a' and 'b' represent elements from two different schemas and 'b' is embedded inside 'a', the proper syntax for schema identification is:

```
<a xmlns="urn:newsml:iptc.org:a"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:newsml:iptc.org:a  a.xsd   urn:newsml:iptc.org:b
   b.xsd">
  <b xmlns="urn:newsml:iptc.org:b">
        <bb>ffff</bb>
```

```
    </b>
  </a>
```

Example using NewsML v1.2 as the container schema:

```
<NewsML xmlns="urn:newsml:iptc.org:20031010:NewsML"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:newsml:iptc.org:20031010:NewsML
http://www.provider.com/xsd/NewsML_1.2.xsd   urn:newsml:iptc.org:specialized
http://www.provider.com/xsd/specialized.xsd ">
    ...
    <NewsItem>
        ...
        <NewsComponent>
            ...
            <ContentItem>
                <DataContent>
                    <specialized xmlns="urn:newsml:iptc.org:specialized">
                        ...
                    </specialized >
                </DataContent>
            </ContentItem>
        </NewsComponent>
    </NewsItem>
</NewsML>
```

10.5 Versioning of the NewsML standard

At present NewsML exists in version 1.0, 1.1 and 1.2. The updates are considered as minor versions, and they are backward compatible with version 1.0: new elements and attributes have been added as requirements became more explicit, but no elements or attributes have been suppressed, so a version 1.2 compliant processor can process all version 1.0 and 1.1 compliant instances.

Forward compatibility: users who receive a version later than version 1.0 may not be able to handle any new elements and attributes with their version 1.0 compliant software. These systems should ignore any unrecognised elements and attributes, and elements contained by the unrecognised elements, without causing a system processing failure (*Must Ignore All* rule). If systems are not built with this protection then they must be capable of recognising a document type other than version 1.0 and discard these documents.

The next major version will be called 2.0, and this version may break the compatibility chain. Developers are advised to check with the IPTC web site for newer documentation.

Subsequent to the release of version 1.0 of the NewsML DTD it was decided that future release should include a **Version** attribute on the NewsML element. So that NewsML 1.0 conformant documents remain compatible with subsequent DTDs this attribute is optional but must always be included in NewsML instances when they are conformant with a DTD subsequent to 1.0 – absence of the version attribute means that the document is compatible with NewsML 1.0.

Related readings:

- W3C - Versioning XML Languages: http://www.w3.org/2001/tag/doc/versioning-20031003
- FOLDOC definition of backward compatibility: http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?backward+compatibility

Creators: L. Le Meur
Main contributors: Jo Rabin