



*IPTC NewsML
News Agency
Implementation
Guidelines*

[Comité International des Télécommunications de Presse](#)

NewsML - User Guideline for News Agencies

Compiled by David Allen, Managing Director IPTC

Copyright IPTC © 2001

Introduction

NewsML is an extensible management format for news information of all types. It consists of a single XML Document Type Definition (DTD) but allows a number of different types of document instances to be valid against the DTD. The document types are NewsML with contained NewsItem and NewsComponents, TopicSets and Catalogs. A Catalog is used by a provider to give information to users on how to locate the various parts of NewsML that are used in the providers services. The TopicSets provide data in controlled vocabularies to populate those parts of NewsML where vocabulary information is required, in particular DescriptiveMetadata. This Guideline provides advice on good practices for News Agencies using NewsML in their services.

Validation

NewsML is intended to be used widely for news and has to exist in a standard form to allow providers and recipients to process the information in a consistent manner. NewsML has been Trademarked by IPTC and the published DTD must be used to validate all instances claiming to be NewsML. This is normally done by declaring the document type as follows:

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20001006:NewsMLv1.0.dtd:1"
"http://www.iptc.org/DTD/NewsMLv1.0.dtd">
```

However, such an approach while allowing validation against the DTD posted on the iptc.org web site also creates an unacceptable burden on the IPTC server and renders the provider and his users at risk from any failure to access this server. The IPTC server is not failsafe and does not form part of a guaranteed 24x7x365 service.

A safer approach would be 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.

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20001006:NewsMLv1.0.dtd:1"
"http://www.provider.org/DTD/NewsMLv1.0.dtd">
```

or

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20001006:NewsMLv1.0.dtd:1" "./DTD/NewsMLv1.0.dtd">
```

Could be suitable ways of indication the SYSTEM locations of a copy of the DTD.

Of course, any reputable provider should ensure that his contract with his users stipulate that NewsML instances will be valid in accordance with the IPTC published DTD and they do not have to validate instances on receipt. Validation can normally be switched of in XML systems when this is required.

Version Changes

At present NewsML exists only in version 1 but as it evolves over time further versions may well be published. The intention is that all future versions will, as far as practical, be backward compatible with version 1. Future versions are likely to have more named elements and attributes as requirements become more explicit and naming, rather than using generic extensions, is considered desirable. In this event users who receive a version later than version 1 may not be able to handle any new elements and attributes with their version 1 systems. Their systems should ignore any unrecognised elements and attributes without causing a system processing failure. If systems are not built with this protection then they must be capable of recognising a document type other than version 1 and discarding the data accordingly. Pending the introduction of any explicit NewsML version identifier attribute the version can be identified by the last character of the PUBLIC IDENTIFIER of the DOCTYPE.

```
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20001006:NewsMLv1.0.dtd:x"
```

Default TopicSets

IPTC

Some of the controlled vocabularies provide data to populate attributes 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. These are NewsItemType, Status, Priority and Subject(Matter,Detail,Qualifier). The IPTC has published 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 extension TopicSets for these element attributes should be included within every NewsItem that employs any of the extended values.

Provider

In order to ensure that users are able to route incoming NewsItems properly, Providers who choose to make use of the NewsEnvelope, need to ensure that the following element attributes within NewsEnvelope are documented in TopicSets provided to their users:- SentFrom, SentTo, NewsService and NewsProduct.

Other TopicSets (eg Role) may be developed, as necessary, to allow users to interpret aspects of the added value a provider makes to his news.

Alerts

An alert requires the minimum of overhead in order not to be delayed in transit and to allow the recipient to grasp the essential information as quickly as possible. The NewsItemType TopicSet has been extended to include the following entry.

```
<Topic Duid="nit09">
  <TopicType Scheme="IptcTopicType" FormalName="NewsItemType"/>
  <FormalName Scheme="IptcNewsItemType">Alert</FormalName>
  <Description xml:lang="en">Preliminary warning message about an important breaking news
  event.</Description>
</Topic>
```

An alert can now be easily indicated by providing a NewsItemType of Alert. The message content may either be included in a NewsLine or as text in a ContentItem depending on the Provider's requirements.

NewsML Compliance

Users of NewsML are licensed only to publish NewsML instances that are in conformance with the DTD. Validation of correct structure can be made using a validating parser and the published DTD. However, the NewsML specification requires rules to be followed that cannot be validated by a normal XML parser. To assist developers an XSL stylesheet has been developed that may be used to check that prototype instances do meet the NewsML specification. It allows that verification of certain #PCDATA values that must have specific values and/or structure as well as vocabularies and Catalogs needed to ensure that FormalName values are properly identified and references. Extensions using the generic Metadata and Characteristics elements may also be tested. This stylesheet only requires a working XSLT parser such as that built into MSIE5+. It may be downloaded from the www.iptc.org site.

Metadata Inheritance

Catalog

The design of NewsML allows for the insertion of Catalogs and TopicSets at various locations within the NewsML structure. A Catalog is defined as a container for Resource and TopicUse elements. Resource elements map URNs to URLs and indicate default external vocabularies which apply to the FormalNames of certain elements within the subtree that begins with the immediate parent of the Catalog element. Catalogs may be placed within the following elements. NewsML, NewsItem, NewsComponent, AdministrativeMetadata, DescriptiveMetadata, RightsMetadata, Metadata, ContentItem, TopicSet and Topic. Hence, as any Catalog will provide information for finding the

vocabularies appropriate to the FormalName of the child elements of the Catalog's parent element, and it is possible that more than one Catalog may exist in the NewsML structure this could lead to confusion over where vocabularies may be found. The Rule that applies is that elements in the tree always inherit Catalog information from the nearest point above them in the tree. In other words the information in the "closest" Catalog overrides any from more distant Catalogs.

Where a FormalName attribute has associated with it a Vocabulary attribute this provides a pointer to a TopicSet which is the controlled vocabulary that can be used to resolve the meaning of the FormalName. 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.

If there is no Vocabulary attribute, then the controlled vocabulary to be used is located by the following algorithm:

- Proceed to the parent 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 contains an Xpath pattern that is matched by the current element. If so, then the controlled vocabulary is the resource identified by that Resource element.
- If the parent does not meet the above condition, proceed to its parent and check the same condition.
- Continue until a vocabulary is found, or no further parent elements are available because the root element has been reached and it too fails to meet the condition. This is an error in the NewsML document.

TopicSet

TopicSets may also be located at different levels in NewsML. The following elements may contain TopicSets: NewsML, NewsItem and NewsComponent. When located within any of these elements they are termed Local Vocabularies. A TopicSet may be pointed to through the use of the Vocabulary attribute. If present it provides a pointer to a TopicSet that is the controlled vocabulary that can be used to resolve the meaning of the FormalName. The value of the Vocabulary attribute is an http URL or a NewsML URN (for external Vocabularies), or the # character followed by the value of the Duid attribute of a TopicSet (Local Vocabulary) in the current document. As the Vocabulary pointer can only identify a specific TopicSet, the location of the TopicSet is not relevant to inheritance. Although a VocabularyPointer could be used to override a default vocabulary pointed to by a Catalog.

The FormalName element (of the Topic) may have a Scheme attribute to indicate that it belongs to a particular naming scheme. It is an error for there to exist two Topics in the same TopicSet that have the same FormalName with the same Scheme attribute. The Scheme attribute, if present, serves to distinguish which of possibly multiple naming schemes in the controlled vocabulary is the one that governs this FormalName. For a match to be obtained within the controlled vocabulary, the rule is that the FormalName and the Scheme must both match. If there is no Scheme attribute on the current element, the match will be to an item in the vocabulary that has the current FormalName and no Scheme. If there is a Scheme attribute on the current element, then both the FormalName and the Scheme in the controlled vocabulary must match. In order to avoid problems it is good practice to ensure that the Scheme attribute is always used and a match is made in a positive sense using both the FormalName and the Scheme values of a Topic in a TopicSet.

Local Topic additions may be made to external default controlled vocabularies using the TopicSet Ref element. This is a pointer to a TopicSet that is to be merged with the current one. The 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 included by reference within the current local TopicSet. When this merging results in there existing two FormalName grandchildren of the same TopicSet that have the same content and the same Scheme attribute value, then the Topics whose children these are, in fact are the same Topic, and are deemed to be merged. The merging of Topics need not be performed physically by the system, but the meaning of the data is exactly the same as if the merging were actually performed. Merging two Topics consists of creating a single Topic that contains all the children of both, and eliminating duplicates. This method allows for additional Schemes to be added to Topics in default external TopicSets.

Extension Mechanisms

NewsML may be legitimately extended in the following ways:

Structural Extensions

Within the designated metadata (AdministrativeMetadata, RightsMetadata and DescriptiveMetadata) additional features may be added using the Property element.

New Metatadata may be added by declaring a MetaDataType and then adding the required number of Property elements.

Default controlled vocabularies may be extended by adding additional entries with new Schemes inside existing TopicSets.

User defined TopicSets may be referenced from a NewsML instance through the user's Catalog. It is helpful if both the containing Topicset and its child Topics have the same TopicType. Thus only one corresponding entry is needed in the TopicType TopicSet.

Content Inclusions

NewsML may include content inline or externally by reference.

Where content is inline within ContentItem and DataContent validation can be maintained by either:

Declaring a local subset for the NewsML DTD within the document instance (allows additional elements and attributes to be part of the DataContent ANY content model.

Including a reference to an external DTD by means of an entity reference in the document instance declaration.

Through the use of an additional namespace attribute within the NewsML element.

This allows external content structures to be embedded within NewsML but it is not permitted to modify the NewsML document other than within the DataContent. To do so would be a breach of the NewsML licence and a violation of the NewsML trademark. IPTC will take appropriate action to protect its rights in the event of any discovered breach of its NewsML licence or trademark.

The Property element

The Property element is used to assert the value of some property on Characteristics, AdministrativeMetadata, DescriptiveMetadata, RightsMetadata, Metadata or Topic. The property must be formally named and may contain sub-properties to handle complex properties.

The Property has a name and either a simple or a complex value consisting of a set of further properties. The Value attribute provides a string representation of the value of a Property. The ValueRef attribute gives a pointer to the value of the Property. This might be a Topic in a TopicSet, or any other piece of data. 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. It is an ERROR condition if the resolved value pointed to by the ValueRef attribute and the Value specified are not the same. The AllowedValues attribute, if present, is a pointer to a controlled vocabulary that delimits the set of allowed values for the property. (The vocabulary does not have to conform to the NewsML specification.) This may be an http URL, or a NewsML URN, or a fragment identifier consisting of a # character followed by the Duid of an element in the current document. The pointer must reference either a Resource element that designates an external controlled vocabulary, or a TopicSet element, that is itself the controlled vocabulary.

Element	Attributes	
Property	FormalName	Value
		ValueRef
	Vocabulary	AllowedValues

The name given to a Property is drawn from a Topic in a TopicSet where the FormalName attribute value is defined. The Vocabulary attribute provides a pointer to the appropriate TopicSet. The Value attribute contains a string representation of the actual value assigned. The ValueRef attribute provides a pointer to the exact Topic in a TopicSet where this value is specified. The AllowedValues attribute provides a pointer to the TopicSet where all the possible values for the Value attribute of this Property are listed.

The attributes Value, ValueRef and AllowedValues may occur in the following combinations:

1. Value. In this case the value assigned is uncontrolled.

```
<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName" >
  <Topic Duid="pcf1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">FCountry</FormalName >
    <Description xml:lang="en">Name of imaginary geo-political entity.</Description >
  </Topic >
</TopicSet >

<Property FormalName="FCountry" Vocabulary="#myprops" Scheme="MyProperty" Value="Narnia"/>
```

2. Value with a corresponding Resource entry in a Catalog. In this case the value assigned is controlled by the TopicSet pointed to by the Resource element Url child.

```
<Catalog>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:topicset.fictional-country:1</Urn >
    <Url>./topicsets/fictional-country.xml</Url >
    <DefaultVocabularyFor Context="Property[@FormalName='FCountry']/@Value"/>
  </Resource >
</Catalog >

<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName" >
  <Topic Duid="pcf1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">FCountry</FormalName >
    <Description xml:lang="en">Name of imaginary geo-political entity.</Description >
  </Topic >
</TopicSet >

<Property FormalName="FCountry" Vocabulary="#myprops" Scheme="MyProperty" Value="Narnia"/>
```

3. Value and ValueRef. The value assigned is that pointed to by the ValueRef pointer. These must be the same value otherwise an ERROR exists.

CORRECT

```
<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName" >
  <Topic Duid="pcf1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">FCountry</FormalName >
    <Description xml:lang="en">Name of imaginary geo-political entity.</Description >
  </Topic >
</TopicSet >

<TopicSet Duid="fcos" Scheme="ExtraTopicType" FormalName="FCountry">
  <Topic Duid="fco1" >
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Narnia</FormalName >
    <Description xml:lang="en">From the novels of C.S Lewis</Description >
  </Topic >
  <Topic Duid="fco2" >
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Lilliput</FormalName >
    <Description xml:lang="en">From Gullivers Travels</Description >
  </Topic >
</TopicSet >
```

```
</TopicSet>
```

```
<Property FormalName="FCountry" Vocabulary="#myprops" Scheme="MyProperty" Value="Narnia"
ValueRef="#fco1"/>
```

ERROR

```
<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName">
  <Topic Duid="pcf1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">FCountry</FormalName>
    <Description xml:lang="en">Name of imaginary geo-political entity.</Description>
  </Topic>
</TopicSet>
```

```
<TopicSet Duid="fcos" Scheme="ExtraTopicType" FormalName="FCountry">
  <Topic Duid="fco1">
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Narnia</FormalName>
    <Description xml:lang="en">From the novels of C.S Lewis</Description>
  </Topic>
  <Topic Duid="fco2">
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Lilliput</FormalName>
    <Description xml:lang="en">From Gullivers Travels</Description>
  </Topic>
</TopicSet>
```

```
<Property FormalName="FCountry" Vocabulary="#myprops" Scheme="MyProperty" Value="Narnia"
ValueRef="#fco2"/>
```

- Value and AllowedValues. The value specified is drawn from the TopicSet pointed to by the AllowedValues attribute. (Normally this would be a local TopicSet and AllowedValues behaves in a similar way to the Vocabulary attribute.)

```
<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName">
  <Topic Duid="pcf1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">FCountry</FormalName>
    <Description xml:lang="en">Name of imaginary geo-political entity.</Description>
  </Topic>
</TopicSet>
```

```
<TopicSet Duid="fcos" Scheme="ExtraTopicType" FormalName="FCountry">
  <Topic Duid="fco1">
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Narnia</FormalName>
    <Description xml:lang="en">From the novels of C.S Lewis</Description>
  </Topic>
  <Topic Duid="fco2">
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Lilliput</FormalName>
    <Description xml:lang="en">From Gullivers Travels</Description>
  </Topic>
</TopicSet>
```

```
<Property FormalName="FCountry" Vocabulary="#myprops" Scheme="MyProperty" Value="Narnia"
AllowedValues="#fcos"/>
```

- ValueRef. This just provides a pointer to the (local) TopicSet Topic child where the value that would be assigned to the Value attribute (if it was used) may be found.

```

<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName">
  <Topic Duid="pcf1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">FCountry</FormalName>
    <Description xml:lang="en">Name of imaginary geo-political entity.</Description>
  </Topic>
</TopicSet>

<TopicSet Duid="fcos" Scheme="ExtraTopicType" FormalName="FCountry">
  <Topic Duid="fco1">
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Narnia</FormalName>
    <Description xml:lang="en">From the novels of C.S Lewis</Description>
  </Topic>
  <Topic Duid="fco2">
    <TopicType Scheme="ExtraTopicType" FormalName="FCountry"/>
    <FormalName>Lilliput</FormalName>
    <Description xml:lang="en">From Gullivers Travels</Description>
  </Topic>
</TopicSet>

<Property FormalName="FCountry" Vocabulary="#myprops" Scheme="MyProperty" ValueRef="#fco1" />

```

Extending AdministrativeMetadata

```

<Resource>
  <Urn>urn:newsml:iptc.org:20001006:topicset.iso-country:1</Urn>
  <Url>./topicsets/iso-country.xml</Url>
  <DefaultVocabularyFor Scheme="ISO3166-alpha3"
Context="AdministrativeMetadata/Property[@FormalName='Country']/@Value"/>
</Resource>

<TopicSet Duid="iptc.entities" Scheme="ExtraTopicType" FormalName="IptcEntity">
  <Comment xml:lang="en">The relevant entities for IPTC publications.</Comment>
  <Topic Duid="ent01">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">IPTC</FormalName>
    <Description xml:lang="en">International Press Telecommunications Council.</Description>
  </Topic>
  <Topic Duid="ent02">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">David Allen</FormalName>
    <Description xml:lang="en">IPTC Managing Director.</Description>
  </Topic>
  <Topic Duid="ent04">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">Hugh Johnstone</FormalName>
    <Description xml:lang="en">IPTC Editor.</Description>
  </Topic>
  <Topic Duid="ent05">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">All NewsML Users</FormalName>
    <Description xml:lang="en">Main audience for web site.</Description>
  </Topic>
  <Topic Duid="ent06">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">IPTC NewsML</FormalName>
    <Description xml:lang="en">Web site service name.</Description>
  </Topic>
  <Topic Duid="ent07">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">WebSite</FormalName>
    <Description xml:lang="en">Publishing vehicle.</Description>
  </Topic>
  <Topic Duid="ent08">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">General</FormalName>

```



```

    <Description xml:lang="en">Of interest to all readers.</Description>
  </Topic>
  <Topic Duid="ent09">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcEntity"/>
    <FormalName Scheme="PubEntities">Feature</FormalName>
    <Description xml:lang="en">A specially prepared article.</Description>
  </Topic>
</TopicSet>

<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName">
  <Topic Duid="pc1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">Country</FormalName>
    <Description xml:lang="en">Name of geo-political entity.</Description>
  </Topic>
  <Topic Duid="pp1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">PropertyOne</FormalName>
    <Description xml:lang="en">First property of my Metadata</Description>
  </Topic>
  <Topic Duid="pp2">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">PropertyTwo</FormalName>
    <Description xml:lang="en">Second property of my Metadata</Description>
  </Topic>
</TopicSet>

<AdministrativeMetadata Duid="Admta1">
  <FileName>webpage.xml</FileName>
  <SystemIdentifier>webpage.xml</SystemIdentifier>
  <Provider>
    <Party FormalName="IPTC" Vocabulary="#iptc.entities" Scheme="PubEntities"/>
  </Provider>
  <Creator>
    <Party FormalName="David Allen" Vocabulary="#iptc.entities" Scheme="PubEntities"/>
  </Creator>
  <Source>
    <Party FormalName="IPTC" Vocabulary="#iptc.entities" Scheme="PubEntities"/>
  </Source>
  <Contributor>
    <Party FormalName="Hugh Johnstone" Vocabulary="#iptc.entities"
Scheme="PubEntities"/>
  </Contributor>
  <Property FormalName="Country" Vocabulary="#myprops" Scheme="MyProperty"
Value="JPN"/>
</AdministrativeMetadata>

```

Extending DescriptiveMetadata

This example illustrates how the an additional Classifier may be added to Descriptive Metadata

```

<Resource>
  <Urn>urn:newsml:iptc.org:20001006:topicset.classifier:1</Urn>
  <Urb>#iptc.concept</Urb>
  <DefaultVocabularyFor Scheme="IptcClassifiers"
Context="DescriptiveMetadata/Property[@FormalName='Classifier']/@Value"/>
</Resource>
<Resource>
  <Urn>urn:newsml:iptc.org:20001006:topicset.concept:1</Urn>
  <Urb>#myconcepts</Urb>
  <DefaultVocabularyFor Scheme="MyConcept"
Context="DescriptiveMetadata//Property[@FormalName='Concept']/@Value"/>
</Resource>
<Resource>
  <Urn>urn:newsml:iptc.org:20001006:topicset.brico:1</Urn>
  <Urb>#bricolist</Urb>
  <DefaultVocabularyFor Scheme="Brico"
Context="DescriptiveMetadata//Property[@FormalName='Brico']/@Value"/>
</Resource>

```

```

<TopicSet Duid="iptc.classifier" Scheme="ExtraTopicType" FormalName="IptcClassifier">
  <Comment xml:lang="en">A Topic inspired by Ken Haas and his "brico" project. </Comment>
  <Topic Duid="clas01">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcClassifier"/>
    <FormalName Scheme="IptcClassifiers">Classifier</FormalName>
    <Description xml:lang="en">The entire grouping of document descriptors that could be applied to
an entire document or a document subsection.</Description>
  </Topic>
</TopicSet>
<TopicSet Duid="iptc.concept" Scheme="ExtraTopicType" FormalName="IptcConcept">
  <Topic Duid="ccpt01">
    <TopicType Scheme="ExtraTopicType" FormalName="IptcConcept"/>
    <FormalName Scheme="IptcClass">Concept</FormalName>
    <Description xml:lang="en">A disambiguated keyword describing a meaning that is associated
with the content it encloses or refers to.</Description>
  </Topic>
</TopicSet>
<TopicSet Duid="myconcepts" Scheme="ExtraTopicType" FormalName="ConceptName">
  <Topic Duid="cc1">
    <TopicType Scheme="ExtraTopicType" FormalName="ConceptName"/>
    <FormalName Scheme="MyConcept">Brico</FormalName>
    <Description xml:lang="en">A Ken Haas project.</Description>
  </Topic>
</TopicSet>
<TopicSet Duid="bricolist" Scheme="ExtraTopicType" FormalName="BricoName">
  <Topic Duid="bc1">
    <TopicType Scheme="ExtraTopicType" FormalName="BricoName"/>
    <FormalName Scheme="Brico">Pasta</FormalName>
    <Description xml:lang="en">Italian food.</Description>
  </Topic>
</TopicSet>

<DescriptiveMetadata Duid="Ddmta1">
  .....
  <Property FormalName="Classifier" Vocabulary="#iptc.classifier" Scheme="IptcClassifiers"
Value="Concept" AllowedValues="#iptc.concept">
    <Property FormalName="Concept" Vocabulary="#iptc.concept" Scheme="IptcClass"
Value="Brico" AllowedValues="#myconcepts">
      <Property FormalName="Brico" Vocabulary="#myconcepts" Scheme="MyConcept"
Value="Pasta" AllowedValues="#bricolist"/>
    </Property>
  </Property>
</DescriptiveMetadata>

```

Adding a New Metadata Type

1. Declare new TopicTypes

```

<TopicSet Duid="newsmltopictypes" Scheme="IptcTopicType" FormalName="TopicType">
  <TopicSetRef TopicSet="urn:newsml:iptc.org:20001006:topicset.iptc-topictype:1"/>
  <Topic Duid="MyTopic.Meta">
    <TopicType Scheme="IptcTopicType" FormalName="TopicType"/>
    <FormalName Scheme="ExtraTopicType">MetaName</FormalName>
    <Description xml:lang="en">Naming of additional Metadata types</Description>
  </Topic>
  <Topic Duid="MyTopic.PropN">
    <TopicType Scheme="IptcTopicType" FormalName="TopicType"/>
    <FormalName Scheme="ExtraTopicType">PropertyName</FormalName>
    <Description xml:lang="en">Naming of additional Metadata Properties</Description>
  </Topic>
  <Topic Duid="MyTopic.PropV">
    <TopicType Scheme="IptcTopicType" FormalName="TopicType"/>
    <FormalName Scheme="ExtraTopicType">PropertyValue</FormalName>
    <Description xml:lang="en">Naming of additional Metadata Property values</Description>
  </Topic>
</TopicSet>

```

2. Declare Metadata names

```

<TopicSet Duid="mymetadata" Scheme="ExtraTopicType" FormalName="MetaName">
  <Topic Duid="mm1">
    <TopicType Scheme="ExtraTopicType" FormalName="MetaName"/>
    <FormalName Scheme="MyMeta">MetaTypeOne</FormalName>
    <Description xml:lang="en">My first extra Metadata type</Description>
  </Topic>
  <Topic Duid="mm2">
    <TopicType Scheme="ExtraTopicType" FormalName="MetaName"/>
    <FormalName Scheme="MyMeta">MetaTypeTwo</FormalName>
    <Description xml:lang="en">My second extra Metadata type</Description>
  </Topic>
</TopicSet>

```

3. Declare Property names

```

<TopicSet Duid="myprops" Scheme="ExtraTopicType" FormalName="PropertyName">
  <Topic Duid="pc1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">Country</FormalName>
    <Description xml:lang="en">Name of geo-political entity.</Description>
  </Topic>
  <Topic Duid="pp1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">PropertyOne</FormalName>
    <Description xml:lang="en">First property of my Metadata</Description>
  </Topic>
  <Topic Duid="pp2">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyName"/>
    <FormalName Scheme="MyProperty">PropertyTwo</FormalName>
    <Description xml:lang="en">Second property of my Metadata</Description>
  </Topic>
</TopicSet>

```

4. Declare Property values.

```

<TopicSet Duid="propone" Scheme="ExtraTopicType" FormalName="PropertyValue">
  <Topic Duid="propone1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyValue"/>
    <FormalName Scheme="PropertyOneValueScheme">P1value1</FormalName>
    <Description xml:lang="en">First value for property1 of my Metadata</Description>
  </Topic>
  <Topic Duid="propone2">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyValue"/>
    <FormalName Scheme="PropertyOneValueScheme">P1value2</FormalName>
    <Description xml:lang="en">Second value for property1 of my Metadata</Description>
  </Topic>
  <Topic Duid="proptwo1">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyValue"/>
    <FormalName Scheme="PropertyTwoValueScheme">P2value1</FormalName>
    <Description xml:lang="en">First value for property2 of my Metadata</Description>
  </Topic>
  <Topic Duid="proptwo2">
    <TopicType Scheme="ExtraTopicType" FormalName="PropertyValue"/>
    <FormalName Scheme="PropertyTwoValueScheme">P2value2</FormalName>
    <Description xml:lang="en">Second value for property2 of my Metadata</Description>
  </Topic>
</TopicSet>

```

5. Insert Metadata within NewsComponent.

```

<Metadata Duid="mext1">

```

```

    <MetadataType FormalName="MetaTypeOne" Vocabulary="#mymetadata"
Scheme="MyMeta"/>
    <Property FormalName="PropertyOne" Vocabulary="#myprops" Scheme="MyProperty"
Value="P1value1" AllowedValues="#propone"/>
    <Property FormalName="PropertyTwo" Vocabulary="#myprops" Scheme="MyProperty"
ValueRef="#proptwo1"/>
</Metadata>

```

Content Inclusion Options (In order of preference)

- 1=. By reference, using a hyperlink.
- 1=. Use a namespace and to declare the namespace within the content item.
3. Insert a local DOCTYPE declaration at the head of the instance declaring the elements that are being used within the content text.
4. Use BINHEX coding which again makes the content entirely opaque to the NewsML processor.
5. Keep the text in whatever form it is and insert it within a CDATA section so that it is completely opaque to the NewsML processor.
6. Insert plain text (ensuring there are no "mark-up" characters present).

Examples

method 1: by reference

```
<ContentItem Href="<http://www.me.com/story-body.nitf>" />
```

method 2: using XML Namespaces

(has some implementation support)

```

<ContentItem>
<DataContent>
<nitf xmlns="urn:nitf:iptc.org:20010419:NITF:1">
<body>
<p>Today, Clinton visited...</p>
<p><person>Al Gore</person> also
attended the...</p>
</body>
</nitf>
</DataContent>
</ContentItem>

```

method 3:by declaring the NITF DTD within NewsML OR referring to the NITF DTD as an external entity for this document instance (in the internal subset)

```

<ContentItem>
<DataContent>
<nitf>
<body>
<p>Today, Clinton visited...</p>
<p><person>Al Gore</person> also
attended the...</p>
</body>

```

```
</nitf>  
</DataContent>  
</ContentItem>
```

method 4: binhex the whole data content - without using CDATA

```
<ContentItem>  
<Encoding notation="binhex">  
<DataContent>LJASDLFJSL</DataContent>  
</Encoding>  
</ContentItem>
```

method 5: by CDATA

```
<ContentItem>  
<DataContent>  
<![CDATA[  
<nitf>. . . .</nitf>  
]]>  
</DataContent>  
</ContentItem>
```

method 6: plain text (excluding markup symbols)

```
<ContentItem>  
<DataContent>  
The quick brown fox jumped over the lazy dog.  
</DataContent>  
</ContentItem>_
```