

Detailed Explanation of
NewsML TopicSets

By NSK(Japan Newspaper Association)
5 November 2001

This document provides a detailed explanation of NewsML TopicSets written by NSK (Nihon Shimbun Kyokai = Japan Newspaper Association) on the basis of NewsML *specifications*.

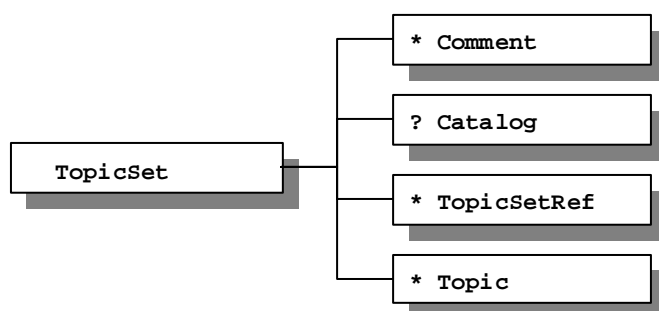
Contents

Contents	2
TopicSet element	3
Controlled vocabulary	3
Place to specify TopicSet element	4
Topic element	5
TopicSetRef element	7
Explanation of Sample TopicSet	8
TopicType element	9
FormalName element	10

TopicSet element

A **TopicSet** is a collection of **Topics**. Note that NewsML has **Topic elements** and **Topic attributes**. The meaning of the **TopicSet** is specified by the **FormalName attributes** (**FormalName attributes** are not shown in the IPTC examples that follow, but they are required for the actual **TopicSet**). The **FormalName attribute** does not have much significance.

A **TopicSet** has **Topic subelements** and several auxiliary **subelements** (one or more **Comment** and **TopicSetRef** and one **Catalog subelement**). A **TopicSet** is a set of declarations. To use a **Topic**, an appropriate **Topic** in the **TopicSet** should be pointed in proper **elements**.



TopicSets can be used for two purposes. The one is to express a set of real-world things (topics). This set may be people, companies, or any other kind of thing that is deemed to be of significance. The other purpose is to use the set as a *controlled vocabulary*.

*** The controlled vocabulary is included in real-world things (topics) in the specifications of NewsML, but this document explains the controlled vocabulary separately.**

Controlled vocabulary

A *controlled vocabulary* is a vocabulary controlled to indicate a unique **Topic** in a **TopicSet** using a combination of the **Scheme attribute** and **FormalName attribute**. In other words, it is a **Topic** that has a **FormalName subelement**.

In a *controlled vocabulary*, a **Topic** corresponds to a candidate for the value to be used, and a **TopicSet** corresponds to a set (*vocabulary*) of candidates for the values to be used. In a *controlled vocabulary*, the number of the **TopicType subelement** is usually one.

Also, for example, when real-world things (topics) are managed with external **TopicSet** files, a **FormalName subelement** may be assigned so that it can be handled as a *controlled vocabulary*. However, the *elements* that point to real-world things (topics) cannot use the **FormalName attribute**. A *controlled vocabulary* is used by such *elements* that point to a **Topic** with the **FormalName attribute**.

Place to specify TopicSet element

TopicSet can be specified at several places in the NewsML documents. When it is specified in the **NewsML** and **NewsComponent** *elements*, it is used as the **TopicSet** that is effective only in each sub-tree. When it is specified in **NewsItem** *elements*, it is used to create **TopicSet** files referenced externally. The NewsML file created in this way is typically used as a *controlled vocabulary*, because the **Topic** attribute pointing to the **Topic** that is not a *controlled vocabulary* exists in **TopicUse**, **TopicOccurrence**, or **Party** *elements*, but its reference scope is limited to the same document.

Place to specify **TopicSet** *element* (1)

The following example shows where **TopicSet** *element* can be specified in a typical NewsML file (which **NewsComponent** is included).

```
<NewsML>
  <Catalog>
    ...
  </Catalog>
  <TopicSet ...>
    ...
</TopicSet>
<NewsItem>
  <NewsManagement>
    <NewsItemType FormalName="News" />
  </NewsManagement>
  <NewsComponent>
    <TopicSet ...>
      ...
    </TopicSet>
  </NewsComponent>
</NewsItem>
</NewsML>
```

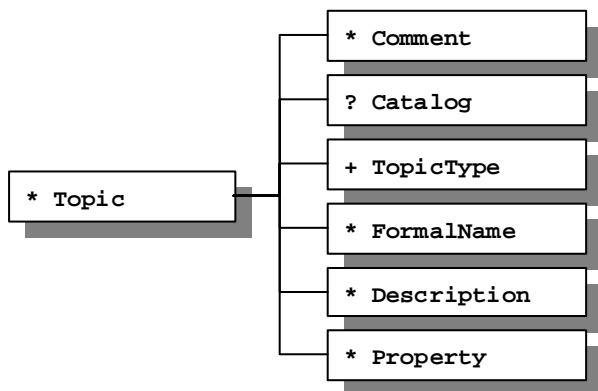
Place to specify **TopicSet** *element* (2)

The following example shows where **TopicSet** *element* can be specified in **TopicSet** file that is referenced externally.

```
<NewsML>
  <NewsItem>
    <NewsManagement>
      <NewsItemType FormalName="TopicSet" />
    </NewsManagement>
    <TopicSet>
      ...
    </TopicSet>
  </NewsItem>
</NewsML>
```

Topic element

A **Topic element** is a reference to one of the real-world things (topics) or one of the items in a *controlled vocabulary*. **Topic elements** may also have a **Details attribute**, which is a pointer, in the form of a URL or URN, to the additional information about the **topic**. The **TopicType subelement** indicates types of **Topic**. Explanation of **Topics** is specified in the **Description subelement**. If **FormalName subelement** exists, it can be used as a *controlled vocabulary*. **Topic elements** also include auxiliary *subelements* (one or more **Comment** and **Property**, and one **Catalog subelement**).



Topic elements are classified into two types, one having **FormalName** and the other not having **FormalName**. Note that NewsML has **FormalName elements** and **FormalName attributes**.

Topic elements that do not have **FormalName subelements** refer to real-world things (topics). They may often have more than one **TopicType subelement**.

Real-world things (topics) are specified to the **Topic attribute** included in the **TopicUse**, **TopicOccurrence**, and **Party elements** with a pointer consisting of a # sign followed by the value of **Duid** of the corresponding **Topic**.

Example of specifying real-world things (topics)

```
<NewsComponent>
  <Catalog>
    <TopicUse Topic="#topic01" Context="DescriptiveMetadata">
  </Catalog>
  <TopicSet FormalName="LocalVocabulary">
    <Topic Duid="topic01">
      <TopicType FormalName="Event" />
      <Description> ... </Description>
    </Topic>
  </TopicSet>
  <DescriptiveMetadata>
    <TopicOccurrence Topic="#topic01" />
    ...
  </DescriptiveMetadata>
```

To assign one value of a *controlled vocabulary*, specify the contents of **FormalName subelement** of the corresponding **Topic** to the **FormalName attribute** included in the object *elements*. You can use *naming scheme* to specify **Topics** more exactly. To use *naming scheme*, specify the value of the **Scheme attribute** included in the **FormalName subelement** of the corresponding **Topic** to the **Scheme attribute** of the object *elements*.

Example of specifying a *controlled vocabulary*

```
<NewsComponent>
  <TopicSet Duid="LocalTopicSet" FormalName="LocalVocabulary">
    <Topic Duid="topic.nskcontentprice">
      <TopicType FormalName="MetadataType" Scheme="IptcTopicType" />
      <FormalName Scheme="NskMetadataType">NSK:ContentPrice</FormalName>
      <Description xml:lang="ja">Content Price</Description>
    </Topic>
    <Topic Duid="topic.nskprice">
      <TopicType FormalName="Property" Scheme="IptcTopicType" />
      <FormalName Scheme="NskProperty">NSK:PriceYEN</FormalName>
      <Description xml:lang="ja">Price (Yen)</Description>
    </Topic>
  </TopicSet>
  <Metadata>
    <Catalog>
      <Resource>
        <Url>#LocalTopicSet</Url>
        <DefaultVocabularyFor Context="MetadataType" />
        <DefaultVocabularyFor Context="Property" />
      </Resource>
    </Catalog>
    <!-- Content Price Information-->
    <MetadataType FormalName="NSK:ContentPrice" />
    <!-- Content Price (Yen) is 1000 -->
    <Property FormalName="NSK:PriceYEN" Value="1000" />
  </Meatadata>
  ...
</NewsComponent>
```

TopicSetRef element

To use existing **TopicSet** as if it is included in new **TopicSet**, **TopicSetRef** *element* can be specified to include all existing **Topics** contained in the new **TopicSet**. For this purpose, a pointer to the **TopicSet** having **Topics** to include is specified as the **TopicSet** *attribute* of the **TopicSetRef** *element*. This pointer is either an http *URL* or a NewsML *URN* identifying an internal or external **TopicSet**, or the *fragment identifier* consisting of a # sign followed by the value of **Duid** *attribute* of a **TopicSet** in the current document.

Example of adding a **Topic** using **TopicSetRef**

The following example adds NSK (Nihon Shimbun Kyokai) to IPTC Provider.

```
<NewsComponent>
  <Catalog>
    <Resource>
      <Url>#LocalTopicSet</Url>
      <DefaultVocabularyFor Context="Party" />
    </Resource>
  </Catalog>
  <TopicSet Duid="LocalTopicSet" FormalName="LocalVocabulary">
    <TopicSetRef
      TopicSet="urn:newsml:iptc.org:20001006:topicset.iptc-provider"/>
    <Topic Duid="NskProvider.NSK">
      <TopicType FormalName="Provider" Scheme="IptcTopicType" />
      <FormalName Scheme="NskTiffServiceId">NSK</FormalName>
      <Description xml:lang="ja"> Nihon Shimbun Kyokai </Description>
    </Topic>
  </TopicSet>
  <AdministrativeMetadata>
    <Provider>
      <Party FormalName="NSK" />
    </Provider>
    <Creator>
      <Party FormalName="IPTC" />
    </Creator>
  </AdministrativeMetadata>
  ...
</NewsComponent>
```

TopicSetRef *elements* can also be used to extend **Topics** that are already defined. If the same combinations of the **FormalName** and **Scheme** are present in the **Topic** to be included by the **TopicSetRef** *element* and in the **Topic** of a new **TopicSet**, these two **Topics** indicate the same item. In such cases, these two **Topics** 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.

Example of merging **Topics** using **TopicSetRef**

The following example adds NSK (Nihon Shimbun Kyokai) and KYODO NEWS to IPTC Provider. KYODO NEWS uses the same topic KYODO as already defined by IPTC. This means that KYODO Topic is treated as KYODO NEWS. (In other words, KYODO NEWS is expected to be handled the same way as KYODO.)

```
<NewsComponent>
  <Catalog>
    <Resource>
      <Url>#LocalTopicSet" </Url>
      <DefaultVocabularyFor Context="Party" />
    </Resource>
  </Catalog>
  <TopicSet Duid="LocalTopicSet">
    <TopicSetRef
```

```

    TopicSet="urn:newsmml:iptc.org:20001006:topicset.iptc-provider"/>
<Topic Duid="IptcProvider.KYODO">
  <TopicType FormalName="Provider" Scheme="IptcTopicType" />
  <FormalName Scheme="NskTiffServiceId">KYODO NEWS</FormalName>
  <FormalName Scheme="IptcProvider">KYODO</FormalName>
  <Description xml:lang="ja"> KYODO NEWS </Description>
</Topic>
<Topic Duid="NskProvider.NSK">
  <TopicType FormalName="Provider" Scheme="IptcTopicType" />
  <FormalName Scheme="NskTiffServiceId">NSK</FormalName>
  <Description xml:lang="ja"> Nihon Shimbun Kyokai </Description>
</Topic>
</TopicSet>
...
</NewsComponent>

```

Explanation of Sample TopicSet

To point the sample **TopicSet** shown above, the following statement is used.

(1) To point the **Topic** with the **FormalName attribute**

```
<Party FormalName="Column B" Scheme="Column C" Vocabulary="#LocalTopicSet"/>
```

(2) To point the **Topic** with the **Topic attribute**

```
<TopicOccurrence Topic="#Column A"/>
```

(Usually, Provider **Topic** is not pointed with **TopicOccurrence**)

Column A Topic/@Duid	Column B Topic/FormalName	Column C Topic/FormalName/@Scheme	Column D Topic/TopicType	Explanator
NskProvider.NSK	NSK	NskTiffServiceId	Provider	
IptcProvider.KYODO	KYODO NEWS	NskTiffServiceId	Provider	Same Topic
	KYODO	IptcProvider	Provider	Same Topic

TopicType element

A **Topic** has one or more **TopicType** *subelements*. The **TopicType** *element* designates types of **Topic**. A **Topic** may have many **TopicType** *subelements*. Types of **Topic** are designated by setting the value of the *controlled vocabulary* prepared for the classification (=type) of **Topic** to the **FormalName** *attribute* of the **TopicType** *element*.

Example of TopicType

```
<Topic Duid="topic.nskstdnewsinfomation">
  <TopicType FormalName="MetadataType" Scheme="IptcTopicType"/>
  <FormalName Scheme="NskMetadataType">NskMetadata</FormalName>
  <Description xml:lang="ja">NSK extension meta data</Description>
</Topic>
```

TopicSet file *urn:newsml:iptc.org:20001006:topicset.iptc-topictype* that lists the **TopicType** of IPTC, or user **TopicSet** file into which foregoing **TopicSet** file is merged by **TopicSetRef**, is used for the *controlled vocabulary* prepared for the classification (= type) of **Topic**. The type of **Topic** in this *controlled vocabulary* is "**TopicType**".

Excerpts from urn:newsml:iptc.org:20001006:topicset.iptc-topictype:1

```
<TopicSet Duid="newsmltopictypes" FormalName="TopicType">
  ...
  <!--Type of TopicDefining "MediaType"-->
  <Topic Duid="TopicTypes.NewsML.MediaType">
    <TopicType FormalName="TopicType"/>
    <FormalName Scheme="IptcTopicType">MediaType</FormalName>
  </Topic>
  ...
```

A certain degree of automatic processing can be expected if the value defined by IPTC is used as the type of **Topic**. For example, it is possible to create a system that automatically registers **Topics** belonging to the type "**Person**" to a *person database*, and "**Event**" to an *event database*.

The *controlled vocabulary* can indicate a common control if the value defined by IPTC for the type of **Topic** and **FormalName** is used. Examples of this are the **Topics** that have "**Status**" or "**NewsItemType**" for the type of **Topic**.

Local meaning should not be given to **FormalName** of IPTC. To specify local meaning, add a new **Topic** and use new **FormalName**. The control for the added **Topic** should be known between a sender and receiver in advance.

FormalName element

The **FormalName** *element* defines the name of **Topic** to be pointed. This *element* can have **Scheme** *attribute*. Note that NewsML includes **FormalName** *elements* and **FormalName** *attributes*.

The **Scheme** *attribute* of the **FormalName** *element* can define any *naming scheme* and assign this **Scheme** to a **FormalName**. Note that *specifications* for the **Scheme** *attribute* of the **FormalName** *element* are different from that for the **Scheme** *attribute* of other *elements*. (The **Scheme** *attribute* of other *elements* specifies existing *naming scheme* assigned by the **Scheme** *attribute* of this **FormalName** *element*, where as the **Scheme** *attribute* of **FormalName** *element* creates a *naming scheme*.)

The combination of contents of the **FormalName** *element* and the value of the **Scheme** *attribute* should be unique in the **TopicSet**. The **Topic** *element* can have more than one **FormalName** *subelement*. As a result, multiple combinations of the values of the **FormalName** and **Scheme** *attributes* may point to a single **Topic**.

Example of multiple FormalName elements

```
<Topic Duid="topic.nsk.kyodo">
  <TopicType FormalName="Provider" Scheme="IptcTopicType" />
  <FormalName Scheme="NskProvider">NSK:Kyodo</FormalName>
  <FormalName Scheme="NskServiceId">KYODO NEWS</FormalName>
  <Description xml:lang="ja"> KYODO NEWS </Description>
</Topic>
```