

日本新聞協会 NewsML レベル 1.2 解説書（第 1.2.1 版）

NewsML サポートチーム作成
(2005.9.15)

はじめに

2003 年 10 月、IPTC (国際新聞電気通信評議会) は NewsML 1 の最新バージョンである NewsML 1.2 を公開しました。2005 年 7 月、社団法人日本新聞協会と財団法人日本規格協会は協力して、NewsML 1.2 を JIS 規格 (日本工業規格 JIS X 7201:2005) としました。これに合わせて、日本新聞協会技術委員会 NewsML サポートチームでは、NewsML 1.2 で新聞・通信社間でデータ交換する場合のガイドラインとして、日本新聞協会 NewsML (以下、NskNewsML) のレベル 1.2 ガイドラインを公開します。この解説書は、IPTC が 2003 年 10 月 10 日に公開した “NewsML Version 1.2 Functional Specification” の日本語訳に、日本新聞協会技術委員会「NewsML チーム」及び「NewsML サポートチーム」が解説を加えたものです。作業としては、日本新聞協会技術委員会 NewsML チームが 2004 年 7 月に作成した NewsML v1.2 機能仕様書日本語訳 (日本新聞協会解説付き) に、NskNewsML レベル 1.2 ガイドライン表、[NskNewsML:1.2 記述形式] などの解説、ツリー図などを NewsML サポートチームが追加したものです。

現在、既に NskNewsML レベル 1 ガイドラインが、新聞・通信社間でデータ交換する場合のガイドラインとして利用されています。2001 年 7 月、NewsML チームは、IPTC が 2000 年 10 月に公開した NewsML の最初のバージョンである NewsML 1.0 をもとに、NskNewsML レベル 1 ガイドラインを公開しました。共同通信社がこのガイドラインに従って配信を行っているなど、日本では多くの新聞、通信社が NskNewsML レベル 1 ガイドラインに従って配信を行っています。レベル 1.2 ガイドラインが公開されても、レベル 1 ガイドラインは有効であり、今後もレベル 1 対応システムを開発することを抑制するものではありません。

このように新聞・通信社間でデータ交換する場合、NskNewsML レベル 1 ガイドラインと NskNewsML レベル 1.2 ガイドラインの 2 つのガイドラインが存在することになります。そこで、この解説書では、最初に、2 つのガイドラインの違いを [NskNewsML レベル 1.2 での仕様変更箇所] に記述しました。また、NskNewsML レベル 1.2 ガイドラインは IPTC NewsML 1.2 に準拠していますが、2 つのガイドラインの互換性をとるために仕様を制限した箇所を、[NskNewsML レベル 1 ガイドラインとの互換性をとるために仕様を制限した箇所] に記述しました。さらに、NskNewsML レベル 1 と NskNewsML レベル 1.2 のデータ交換上の問題についても、[NskNewsML レベル 1 と NskNewsML レベル 1.2 のデータ交換上の問題] に記述しました。

この解説書が NewsML 1.2、NskNewsML レベル 1.2 の理解と普及に役立てば、幸いです。

NskNewsML レベル 1.2 の基本制限事項

- ① NewsML_1.2.dtdを使用し、IPTCの仕様に従って利用する。
 - ② NskNewsMLレベル1.2でDataContent以下にXMLファイルを記述する場合、下記の制限事項があります。
 - ・ 日本語の要素名は使用しない。
 - ・ NewsMLの要素名と同じ要素名を使用しない。
 - ・ DataContent以下で記述するXMLのDTDについては、外部エンティティとして記述する。
 - ③ NewsML文書全体は、検証済みであること。
 - ④ Update要素は使用しない。
-

[NskNewsML レベル 1.2 での仕様変更箇所]

NskNewsML レベル 1.2 における NskNewsML レベル 1 との仕様の変更点は、以下となる。

① DTD が NewsMLv1.0.dtd から NewsML_1.2.dtd に変更になった。

② 以下の 4 つの属性が追加された。

Comment 要素の FormalName 属性

Comment 要素の Vocabulary 属性

Comment 要素の Scheme 属性

Property 要素の AllowedScheme 属性

(ただし、Property そのものが使用停止の箇所は、下位の AllowedScheme も使用停止とする。)

③ NewsML 要素の Version 属性が追加になった。

NskNewsML レベル 1.2 では、必ず、<NewsML Version="1.2">と記述する。IPTC NewsML 1.2 では、NewsML 1.2 で追加された機能を使う場合、必ず、Version="1.2"と記述するが、NewsML 1.0 の機能のみを使用している場合、Version 属性を省略することも可能。しかし、NskNewsML レベル 1.2 では、常に<NewsML Version="1.2">と記述することにした。

④ NskNewsML レベル 1.2 では、NewsProduct 要素を、必ず、<NewsProduct FormalName="NskNewsML:1.2"/>と記述する。NewsProduct 要素は、IPTC NewsML 1.2 では、省略可、繰り返しありとなっている。しかし、NskNewsML レベル 1.2 では、NskNewsML レベル 1 同様 NewsML 文書であることの識別に用い、必須で繰り返しなしとする。

[NskNewsML レベル 1 ガイドラインとの互換性をとるために仕様を制限した箇所]

IPTC NewsML 1.2 (または 1.1) で変更されたが、NskNewsML レベル 1 との互換性のため、仕様変更せず、以下の様に制約する。

① Catalog の Href 属性には、NewsML URN と http URL に加えて、ftp などの URI も利用可能となった。しかし、NskNewsML レベル 1 同様、NewsML URN と http URL のみが利用可能とする。Resource 子要素の Url も同様とする。

② Party 要素の子要素として、Property 要素が追加となったが、使用停止とする。

③ StatusWillChange 要素の複数回の記述が可能となったが、0 回または 1 回の記述とする。

④ DerivedFrom 要素と AssociatedWith 要素の子要素として、FormalName 属性、Vocabulary 属性、Scheme 属性が追加となったが、使用停止とする。

⑤ Creator 要素が繰り返し可能となったが、0 回または 1 回の記述とする。

⑥ Creator 要素と Contributor 要素の子要素として、繰り返し可能な Contribution 要素が追加されたが、使用停止とする。

⑦ DescriptiveMetadata 要素の子要素として、DateLineDate 要素及び Location 要素が追加されたが、使用停止とする。

⑧ ByLineTitle 要素が追加されたが、使用停止とする。

⑨ NewsLines 要素の下位要素の記述順序が任意となったが、NskNewsML レベル 1 と同様固定とする。

⑩ SubHeadLine 要素の記述回数が任意となったが、0 回または 1 回の記述とする。

NskNewsML レベル 1 と NskNewsML レベル 1.2 のデータ交換上の問題

1. XML エンジンレベルの問題

XML エンジンレベルの観点から見ると、NskNewsML レベル 1(NewsML1.0)と NskNewsML レベル 1.2(NewsML1.2)では DTD が異なるため、データ交換を行う場合は双方のシステムがいずれかのガイドラインのレベルに合わせる必要がある。

ただ、複数のシステムから受信するケースなど、データ交換上、システムが両方のガイドラインレベルに対応することが求められることも想定される。XML をハンドリングするミドルウェアにおいては、NewsML1.0 と NewsML1.2 は DTD のファイル名が異なるために、まったく別の DTD として解釈される。両方の DTD に対応するシステムを構築する場合はその点を考慮する必要がある。

2. アプリケーションレベルの問題

NskNewsML レベル 1 と NskNewsML レベル 1.2 の間のデータ交換においては、アプリケーションレベルの問題についても解決する必要がある。

NewsML1.2 は NewsML1.0 に対して上位互換性が保たれているため、レベル 1 のデータの、DTD の宣言、NewsML 要素の Version 属性、NewsProduct 要素の FormalName 属性の記述を変更することで、レベル 1.2 に変換することができる。

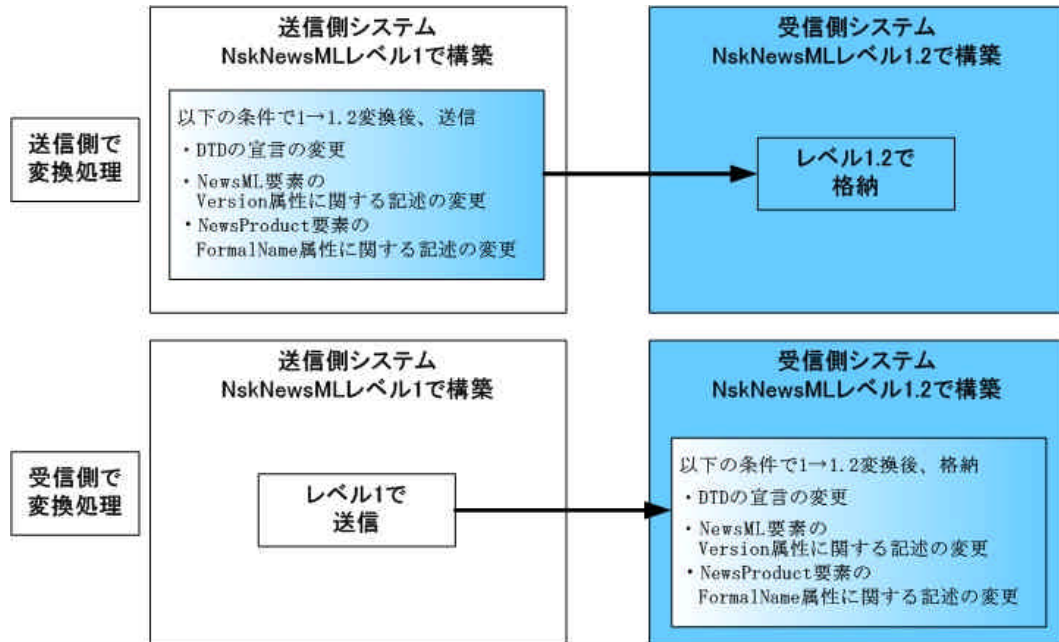
逆に、レベル 1.2 のデータはレベル 1 では表現しきれないため、DTD の宣言、NewsML 要素の Version 属性、NewsProduct 要素の FormalName 属性の記述を変更しただけでは、レベル 1 に変換することはできない。

ただし、レベル 1.2 のデータであっても、レベル 1.2 に追加された機能を使用していた個所を、NewsML の Metadata 下位階層の Property 要素などに割り当てることによって、レベル 1 に変換することはできる。この場合、さらに受信側システムで、その Metadata 下位階層の Property 要素に含めた内容を理解できるように対応してもらう必要がある。

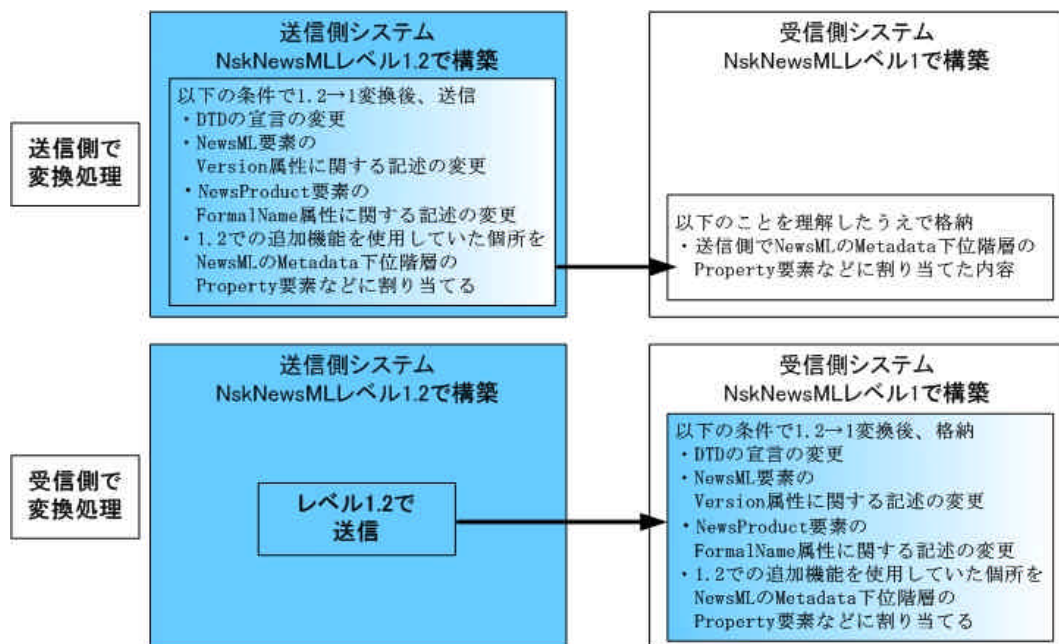
NskNewsML レベル 1 とレベル 1.2 のデータ交換上の概念図

以下は NskNewsML レベル 1 とレベル 1.2 をデータ交換する際の例を図式化したものである。

*NskNewsMLレベル1構築システムからレベル1.2構築システムへ送信する場合 例)



*NskNewsMLレベル1.2構築システムからレベル1構築システムへ送信する場合 例)



目 次

0	利用の手引き	1
1	この文書のステータス	3
2	記述上の規約	3
3	謝辞	3
4	NewsML 概観	5
4.1	NewsML はニュースの交換と管理のための枠組みを提供	5
4.2	NewsML は XML ベース	5
4.3	NewsML はメディア中立	5
5	NewsML の機能	6
5.1	NewsML 文書の構造	6
5.1.1	識別子属性	7
5.2	Catalogs	10
◆	NSK リソース利用方法一覧	31
5.3	TopicSets	33
5.4	NewsEnvelope	45
5.4.1	TransmissionId	47
5.4.2	SentFrom と SentTo	47
5.4.3	DateAndTime	50
5.4.4	NewsService と NewsProduct	51
5.4.5	Priority	52
5.4.6	メタデータの割り当て	53
5.5	NewsItem の構造	55
5.5.1	NewsItem の識別 (NewsIdentifier)	58
5.5.2	非形式識別子	67
5.6	NewsManagement	69
5.6.1	NewsItemType	71
5.6.2	FirstCreated	72
5.6.3	ThisRevisionCreated	72
5.6.4	Status	73
5.6.5	StatusWillChange	75
5.6.6	Urgency	79
5.6.7	RevisionHistory	79
5.6.8	DerivedFrom	80
5.6.9	AssociatedWith	81

5.6.10 Instruction	83
5.6.11 Property	85
5.7 NewsComponent の構造	89
5.7.1 NewsComponents の動きの図解	92
5.7.2 EquivalentList	94
5.7.3 BasisForChoice.....	95
5.7.4 NewsComponent の他の下位要素	98
5.8 ContentItem の構造	99
5.9 メタデータ	105
5.9.1 Administrative Metadata（管理メタデータ）	105
5.9.2 Rights Metadata（権利メタデータ）	110
5.9.3 Descriptive Metadata（記述メタデータ）	114
5.10 NewsLines はメタデータの人間に対する局面を表す	123
5.11 NewsItems の改版	129
5.12 ポインタの使用	136
5.13 NewsML の発展	136
5.14 認証とセキュリティ	136
6 用語	137
7 付録	149
7.1 日本新聞協会 NewsML レベル 1.2 ガイドライン表	149
7.2 地域情報の表現方法について	159
7.3 NewsML 複合ガイドライン	161
7.4 NewsML データ伝送モデルの例	164
7.4.1 NewsML 単体ファイル伝送	164
7.4.2 外部ファイルを伴う複合データ伝送	165

0 利用の手引き

この文書は、[IPTC](#)が2003年10月10日に公開した NewsML Version1.2 Functional Specification を日本語訳したものに、社団法人日本新聞協会技術委員会 NewsML チーム及び NewsML サポートチームが解説を加えたものである。NewsML の理解の手助けをすることを目的として原文のみでは理解が難しい部分について[IPTC](#)の見解に沿った最新の解説を補足している。

なお、機能仕様書の日本語訳には誤りが含まれている恐れがあり、英語版のみを公式な仕様とする。

日本新聞協会「NewsML チーム及び NewsML サポートチーム」が加筆した個所について、この解説書の説明を記述する。

1. 第5章「NewsMLの機能」

[ツリー図]

「NewsMLv1.2.DTD」を元に、該当要素についてのツリーを図にしたもの。要素名の左にある記号は[DTD](#)に規定されているもの。NewsMLv1.1、NewsMLv1.2で追加された要素、属性については、その旨を表記する。

要素名の右にある記号は日本新聞協会（NSK）NewsML レベル1.2で規定したもので、その意味は下記[NskNewsML:1.2記述形式]を参照すること。

[NSK解説]

「機能仕様書 日本語版」をわかりやすく解説したもの。

[NewsML 仕様の変更点：NewsML v1.1]

NewsML v1.0とNewsML v1.1との差分情報。

[NewsML 仕様の変更点：NewsML v1.2]

NewsML v1.1とNewsML v1.2との差分情報。

[コラム]

ポイントとなる解説を“コラム”という形で解説しています。

[NskNewsML:1.2記述形式]

日本新聞協会（NSK）NewsML レベル1.2での要素と属性の利用規定と利用例。

◎…必須：NewsML 文書を交換する上で、なければならないもの。必須の要素や属性のないNewsML 文書はエラーである。

○…省略可：NewsML 文書を交換する上で、省略可能なもの。交換した文書に、この要素や属性があってもなくてもよい。データ交換する当事社間で、利用の可否を決める。

△…規定外：特に規定はしないもの。当事社間で利用してもよいもの。

×…使用停止：使用を停止しているもの。使用停止の要素や属性のあるNewsML 文書はエラーである。

2. 付録

- ・ガイドライン表

各要素や属性について[NskNewsML:1.2記述形式]で規定した必須などの記号を、表形式でまとめたもの。

- ・地域情報の記述方式について

地域情報をMetadataを使って記述する方法について解説したもの。

- ・NewsML複合ガイドライン

複合のガイドラインを解説したもの。これ以外の複合は許可されていない。

- ・NewsMLデータ伝送モデルの例

伝送モデルの例を解説したもの。例であって、これを推奨しているものではない。

1 この文書のステータス

この仕様書は、NewsML1.2 版の文書型定義(DTD)を説明し、補足するものである。

この仕様書の修正は NewsML1.2 版 DTD の注記より優先する。

NewsML の要件 (Requirements) 文書は NewsML が与えるべき能力について述べたものである。現在の仕様書はこのような要求を満たすために採用されてきた技術的手段について説明する。要件は以下のように要約される (かっこの中の R で始まる番号は、NewsML の要件文書の対応する項目の参照番号である)。

NewsML は、コンパクトで(R900)拡張可能かつ柔軟な(R700)ニュースの構造的枠組みであり、[XML](#) と他の適切な技術的標準や仕様に基づく(R1000)。NewsML は電子的なニュースアイテム、ニュースアイテムの集合、ニュースアイテム間の関係、それらに付随するメタデータの表現をサポートしなければならない(R100)。同じ情報が様々な表現で供給されることを見越す(R500)必要があり、任意のメディア・タイプ、フォーマット、言語、エンコードの混在を扱えねばならない(R300,R400)。NewsML はニュースのライフサイクルのすべての段階をサポートせねばならず(R600)、そのライフサイクルにわたってニュースアイテムの改版を許さねばならない(R200)。NewsML はメディアに対して独立だが、テキストを扱う特別の機構を提供することになる(R1100)。NewsML はメタデータとニュースコンテンツ両方に対する認証と署名を提供することになる(R800)。

2 記述上の規約

これ以降の章では、以下のような規約を使う：(訳注 この規約は原文のものであり、日本語訳には必ずしも適用されない)

[下線付きの青い文字](#)は、この文書以外の Web 上のリソースへのハイパーリンクである。

[下線付きの青い太文字](#)は、この文書内でのハイパーリンクである。

斜体 (イタリック) は文書後半の用語集で定義されている術語である。これらの用語には直接その定義を参照できるリンクがついている。MS ワードの "Web" ツールバーにある [青い戻る矢印](#) を押すことで元の位置に戻ることができる。

モノスペース文字は、[XML](#) 要素かその属性の名前、NewsML の文書インスタンスのサンプルか [DTD](#) の断片の記述に用いる。

モノスペースの太文字は説明文の中で [XML](#) 要素あるいは属性の名前を定義するのに用いる。これらの語彙に対する用語集の中にそれらの意味についての短い説明へのリンクが用意される。これらの要素や属性の公式の定義は、NewsML 定義それ自身のなかにも現れる。

青い背景は NewsML DTD からの抜粋の記述に対して使われる。

黄色い背景は、NewsML 文書の断片の実例に対して使われる。

3 謝辞

この仕様書は、国際新聞電気通信評議会([IPTC](#))のメンバーによるチーム作業と外部の人々による協力の成果である。

特に貢献してくれたのは以下の方々である。

この仕様書は Daniel Rivers-Moore(RivCom : 英)によって編集された。作業全体は NewsML 統括委員会によって指揮、監督された。仕様書が承認された時点でのメンバーは、Klaus Sprick (委員長 : Deutsche Press Agentur : 独)、David Allen (IPTC)、James Hartley (Bridge Information Systems : 米)、John Iobst

(米国新聞協会)、Alan Karben (Screaming Media：米)、Laurant Le Meur (AFP：仏)、Irving Levine (ロイター：英)、Kevin Roche (Dow Jones 英)である。この仕様書は、いくつかの IPTC 作業部会、特に NewsML 構造、NewsML メタデータ、NewsML テキスト作業部会との共同作業によるものである。文書による貢献をしたのは IPTC メンバーからは Paul Harman (Press Association：英)、Johan Lindgren (Tidningarnas Telegrambyrå：スウェーデン)、Jo Rabin (ロイター：英)、Tony Rentschler (AP：米)、IPTC の外部からは Martin Bryan (The SGML Centre：英)、Ron Daniel (Metacode：米)、Paul Simmonds (BBC：英)らである。

4 NewsML 概観

NewsML は[XML](#)や他の適切な標準、仕様をもとに、ニュースにコンパクトで、拡張性が高く、柔軟な構造化の枠組みを提供する。電子的なニュースアイテム、ニュースアイテムの集合、それらの間の関係、および関連のメタデータの表現をサポートする。NewsML は同じ情報の複数表現の規定を許し、任意のメディアタイプ、フォーマット、言語、エンコードを混在して使用する。ニュースのライフサイクルのあらゆる場面をサポートし、ニュースアイテムの繰り返しの改版を許す。NewsML はメディア独立だが、テキストを扱うため特別の手法を提供する。NewsML はメタデータとニュースコンテンツ両方の出所を明らかにする。

4.1 NewsML はニュースの交換と管理のための枠組みを提供

NewsML はもともとニュース交換のためのフォーマットとなることを目的としているが、ニュースの蓄積のためのフォーマットとしてや、ネットワーク・コンピューティング環境におけるニュースの作成、編集、管理、発行の補助としても使用される。

4.2 NewsML は XML ベース

NewsML 文書は[XML](#)文書であり、この仕様書の付録 1 に示した NewsML Document Type Definition (文書型定義 = [DTD](#)) に従ったものでなければならない。

すべての[XML](#)文書のように、NewsML 文書は物理的というより論理的なオブジェクトである。NewsML 文書は[XML](#)仕様書で定められたエンティティ参照 ([entity references](#)) あるいは NewsML 文書内のポインタ ([pointer](#)) 機構を使って複数の物理ファイルのコンテンツとして構成されてもよい。

4.3 NewsML はメディア中立

NewsML はメディアタイプ、フォーマット、[news objects](#)のエンコードについて、なにも仮定していない。NewsML 文書はテキスト、ビデオ、オーディオ、グラフィックス、写真、その他のメディア、今後開発されるメディアなど、任意のメディアの組み合わせを含むことができる。

5 NewsML の機能

この章では NewsML 文書構造全体を、そのルート(NewsML) 要素から始めて、各要素 ([element](#))、属性([attribute](#))の構造や目的を説明する。重要な構造は図解例を用意する。

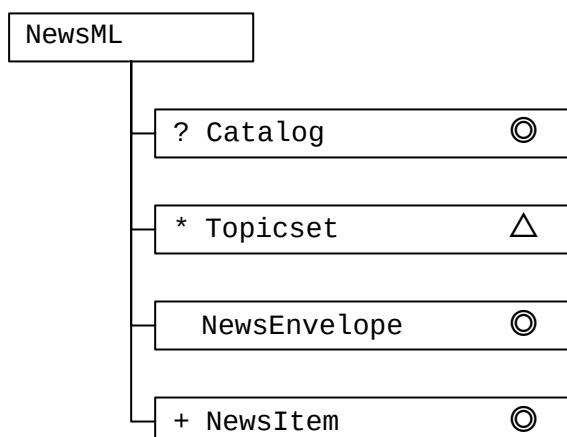
5.1 NewsML 文書の構造

NewsML の要素は完全な NewsML 文書のルート要素である。オプションの [Version](#) 属性は NewsML [DTD](#)や Schemaのバージョンを表し、妥当な文書とするのに使われる。NewsML は[NewsEnvelope](#)と一つ以上の[NewsItem](#)を含まなければならない。NewsML 文書自体の中あるいは NewsML 文書が参照によって含むニュースコンテンツ内に参照される[Topic](#) (または実世界の物事)を含む1個またはそれ以上の[TopicSet](#)要素を含むことができる。また、デフォルトの語彙を識別、指定し、NewsML 文書のどこで、ある[Topic](#)が使われたかを示す[Catalog](#)要素を含んでよい。[Catalog](#) 要素によって[URN](#)を[URL](#)に分解することができ、どの語彙 ([TopicSet](#))がある文脈において与えられた要素タイプにとってデフォルトかを示すことができる。

```
<!ELEMENT NewsML (Catalog?, TopicSet*, (NewsEnvelope , NewsItem+ ))>
<!ATTLIST NewsML %localid
  Version CDATA #IMPLIED>
```

```
<?xml version="1.0"?>
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org: 20031012:NewsMLv1.2.dtd:1"
"http://www.iptc.org/NewsML/DTD/NewsMLv1.2.dtd">
<NewsML Version="1.2">
  <Catalog>
    ...
  </ Catalog >
  <TopicSet>
    ...
  </TopicSet>
  <NewsEnvelope>
    ...
  </NewsEnvelope>
  <NewsItem>
    ...
  </NewsItem>
  <NewsItem>
    ...
  </NewsItem>
</NewsML>
```

[ツリー図]



[NSK 解説]

NewsML 要素は NewsML 文書のルート要素である。オプションの [Version](#) 属性は NewsML DTD や [Scheme](#) のバージョンを表す。[Version](#) 属性は、[XML](#) 文書としてはオプションであるが、NewsML v1.1 以降では必須となっている (Version="1.0" と書いてはいけない)。[Version](#) 属性がオプションであるのは、NewsML v1.1 から追加された属性であるため NewsML v1.1 と互換性を保つためである。また、パラメータエンティティの localid については、「[5.1.1 識別子属性](#)」に説明がされている。NewsML 要素は、デフォルトボキャブラリを定義する [Catalog](#) 要素、文書中にて参照される [Topic](#) を含む [TopicSet](#) 要素、送受信の情報が記述される [NewsEnvelope](#) 要素およびニュースの発行単位である [NewsItem](#) 要素から成る。

[NewsML 仕様の変更点：NewsML v1.1]

NewsML 要素に [Version](#) 属性が追加された。

[NskNewsML:1.2 記述形式]

NewsML/@Version・・◎

NskNewsML レベル 1.2 では、必ず、<NewsML Version="1.2">と記述する。IPTC NewsML 1.2 では、NewsML 1.2 で追加された機能を使う場合、必ず、Version="1.2" と記述するが、NewsML 1.0 の機能のみを使用している場合、Version 属性を省略することも可能。しかし、NskNewsML レベル 1.2 では、常に<NewsML Version="1.2">と記述することにした。

5.1.1 識別子属性

NewsML 文書内の各要素は、[NewsIdentifier](#) とその下位要素以外に、[Duid](#) (ドキュメント・ユニーク識別子) 属性と [Euid](#) (エレメント・ユニーク識別子) 属性の両方あるいはいずれかを、オプションとして持つことができる。これらオプションの目的は同じ文書内や、別の NewsML や [XML](#) 文書でのポインタ参照を可能にすることである。識別子属性の使用により、その文書は世界的に識別される。

5.1.1.1 The "Document-unique" Identifier ("ドキュメント・ユニーク"識別子)

[Duid](#) は [XML](#) の ID 属性規則に従わなければならない。すなわち、[XML](#) 仕様書で定義された名前用文字のみを含み、名前開始文字 (数字であってはいけない) で始めなければならない。その値は NewsML 文書内で唯一でなければならない。

5.1.1.2 The "Element-unique" Identifier (“エレメント・ユニーク” 識別子)

[Euid](#)の値は、同じ要素タイプで同じ親要素を持つ要素間で唯一でなければならない。[Euid](#)属性を使うことで、NewsML 文書ツリーのローカルな枝の文脈の中で、NewsML 要素を識別することができる。これによって、[Duid](#)の唯一性が損なわれるような場合でも（通常ならば新しい[Duid](#)割り当てが必要）各要素のアイデンティティを保持したままで、NewsML 文書のサブツリーをコピーし新たに組み合わせたり、参照によって含んだりすることができる。もし[Euid](#)が各レベルで管理されるなら、たとえば“[Euid](#)が1である[NewsComponent](#)に含まれる[Euid](#)が abc である[ContentItem](#)”というように、[XPointer](#)表現を識別のために使うことができる。そのような識別パターンは、サブツリーを“継ぎはぎ”した後にまで保持される。

```
<!ENTITY % localid " Duid ID #IMPLIED
    Euid CDATA #IMPLIED" >
```

この例では、同じコンテンツが2つの[NewsComponent](#)内で使われている。1番目の[NewsComponent](#)内の[ContentItem](#)は明らかにいくつかのコンテンツ（ここでは...で表わされる）を含む。2番目の[ContentItem](#)は“ツリーを歩き”要求された要素に対する[Euid](#)属性を用いる[XPointer](#)表現を通じ、参照によって1番目の[ContentItem](#)を再利用する。

```
<NewsComponent Duid="a1" Euid="1">
  <ContentItem Euid="abc"> ... </ContentItem>
</NewsComponent>
<NewsComponent Duid="a2" Euid="2">
  <ContentItem Href="#xpointer(//NewsComponent[@Euid='1']/ContentItem[@Euid='abc'])"/>
</NewsComponent>
```

コラム：“[Duid](#)と[Euid](#)の使い分け”

NewsML では、[NewsIdentifier](#)要素とその下位要素及び[TopicUse](#)要素を除くその他すべての要素で、[Duid](#)属性及び[Euid](#)属性が使用可能である。これらの属性は、NewsML 文書内もしくは異なる文書間において、参照する場合の参照先として利用される。[Duid](#)属性と[Euid](#)属性は、類似する名称の属性であるが、使用にあたっては、大きな違いが存在する。以下に、各属性の特徴を記述する。

NewsML の機能仕様書に記述されるように[Duid](#)属性は ID 型であり、記述にあたっては「名前用文字のみを含み、名前開始文字（数字であってはいけない）で始めなければならない」といった、[XML](#)仕様書で定義された用法に従う必要がある。一方[Euid](#)属性は CDATA 型であり、自由な表現が可能である。

まず大きく異なるのが、記述の型式から来る違いである。[Duid](#)属性は ID 型であり、同じ文書内に同じ値を持つ[Duid](#)属性が存在する場合、[XML](#)として検証済みにならない（[XML](#)として間違いである）。一方[Euid](#)属性は CDATA 型であり、同じ親要素を持つ要素間で唯一でなくても、[XML](#)としては検証済みとなる（NewsML としては間違いである）。

	Duid	Euid
記述型式	ID 型	CDATA 型
記述文字	名前開始文字で始める （“123”はエラー）	自由 （“123”も正常）
唯一である必要の範囲	NewsML 文書内	同じ親要素を持つ要素間
唯一であることの保証	XML	NewsML プログラム

以上はXMLとしての側面から見たDuid属性とEuid属性の違いであるが、NewsML としての側面から見た違いについて記述する。

<Duid>

Duid属性は、同じ文書内で唯一性を保証されたものである。そのため、ValueRef属性の記述のように、同じ文書もしくは異なる文書の要素を、Duid属性を記述して直接指定することが可能である。一方、複数の NewsML 文書を1つの NewsML 文書に編集するような場合には、注意が必要である。例えば、2つの NewsML 文書中のNewsItem要素以下を、1つの NewsML 文書中にNewsItem列挙として記述する場合、元の NewsML 文書中で唯一であったDuid属性の値が、1つの NewsML 文書とすることで、唯一性が損なわれる可能性がある。この場合、1つの NewsML 文書とする際に、Duid属性の値の付け直しが必要となる。ここで更に注意が必要なのは、NewsItemごと別の NewsML とする場合でも、Duidの変更によりNewsItemの改版（NewsIdentifier/Revision 要素の値の変更）が必要になることである。これは、NewsItem要素の列挙ではなく、NewsItemRef要素を使用した場合でも同様である。

<Euid>

Euid属性は、同じ親要素を持つ要素間で唯一であることを保証するものであるため、Duid属性に比べ、使用上の制約は少ない。そのため、NewsML 文書の継ぎ接ぎを行う過程において、Duid属性の値の唯一性が損なわれるような場合でも、各要素の独自性を保持したままで、新たな NewsML 文書を生成することが可能となる。一方、Euid属性で管理する要素を指し示すためには、その親要素の特定が必要である。そのため、Euid属性を使用した要素の特定は、異なる文書間のリンクへの使用よりも、文書内の相対的な参照に多く利用されるべきものであると考える。

<補足>

Update要素を使用する場合には、変更の対象となる要素ごとにDuid属性が必須となる。

Duid属性及びEuid属性は、いずれもオプション（省略可）の属性である。これらの属性を使用することで、NewsML の特徴を十分に生かした、参照性の高い NewsML 文書とすることが可能であるが、一方でDuid属性及びEuid属性には、メリット／デメリットが存在することを認識した上で、使用しなければならない。

5.2 Catalogs

NewsML 文書の主要な構造的要素のどれもが、[Resource](#)要素と[TopicUse](#)要素の両方あるいはいずれかを含む[Catalog](#)要素を含むことができる。

それぞれの[Resource](#)要素が、1 個の Uniform Resource Name ([URN](#))と、1 個かそれ以上の Uniform Resource Locator([URL](#))の両方あるいはどちらかを通じて外部のリソースを認識する。それはまた、このリソースが主要な要素の[content](#)の若干数あるいはすべてについて[default vocabulary](#)として働くかどうかを示す。[Urn](#)属性は、NewsML URN が典型的なのだが、そのリソースに対しグローバルな識別子を提供する。[Url](#)下位要素があるとすると、それはそのリソースが見つかり得る場所を指している。[DefaultVocabularyFor](#)要素は、[XPath](#)パターンを内包する。識別されたリソースは、[XPath](#)パターンに合致するすべての要素や属性のために[default vocabulary](#)として働く。[XPath](#)パターンが要素に合致するものであれば、指定された要素の[FormalName](#)属性値である。[XPath](#)パターンが属性に合致するものであれば、指定された属性自体の値である。[XPath](#)パターンは[default vocabulary](#)が適用される文脈を区別するのにある程度、単純であったり複雑であったり得る。

[TopicUse](#)要素は、あるトピックが NewsML 文書内のどこで使われているかを示す。[Topic](#)属性の値は現在の文書において、#記号に、(トピックを記述する) [Topic](#)要素の[Duid](#)属性値を続けて記述したポイントである。[Context](#)属性の値は、現在の[Catalog](#)が適用されているサブツリー内でこのトピックが使われている文脈を示す[XPath](#)パターンである。もし、[Context](#)属性が存在しないのなら、[TopicUse](#)要素は単純に、このトピックがサブツリーのどこかにあると述べているにすぎない。

オプションの[Href](#)属性は、この文書または他の文書内の別の場所にある[Catalog](#)要素へのポイントを提供する。その値は、#記号に、参照された[Catalog](#)要素の[Duid](#)属性値を続けるものからなる。参照された[Catalog](#)が現在の文書内になれば、その[Catalog](#)が出現する文書または[NewsItem](#)を識別する[URI](#)または NewsML URN によって#記号の先に付く。その[Href](#)属性が[Catalog](#)要素上にあれば、要素は空でなければならない。下位要素を含んでいれば、NewsML システムはエラーを示すことになる。

```
<!ELEMENT Catalog (Resource* , TopicUse*)>
<!ATTLIST Catalog %localid;
                Href CDATA #IMPLIED >

<!ELEMENT Resource (Urn? , Url* , DefaultVocabularyFor*)>
<!ATTLIST Resource %localid; >

<!ELEMENT Urn (#PCDATA)>
<!ATTLIST Urn %localid; >

<!ELEMENT Url (#PCDATA)>
<!ATTLIST Url %localid; >

<!ELEMENT DefaultVocabularyFor EMPTY >
<!ATTLIST DefaultVocabularyFor %localid;
                Context CDATA #REQUIRED
                Scheme CDATA #IMPLIED >

<!ELEMENT TopicUse EMPTY >
<!ATTLIST TopicUse Topic CDATA #REQUIRED
                Context CDATA #IMPLIED >
```

次の例は、単一の[Resource](#)と単一の[TopicUse](#)からなる[Catalog](#)を示す。[Resource](#)要素は、IPTC Confidence topic set の第1版のコピーが、[IPTC](#)ウェブサイト上の特定の[URL](#)で見られ、[Confidence](#)属性のための[default vocabulary](#)として働くことを示す。[TopicUse](#)要素は、[Duid](#)属性値がperson1である[Topic](#)が、[DescriptiveMetadata](#)要素の文脈内で使われることを示す。この[Topic](#)は現在の文書内で発生しなければならない。この例では、TopicがIPTC Topic Type vocabularyで定義されるPersonタイプであり、英語でDavid Allen, Managing Director of IPTCと記述していることを宣言している。

```
<Catalog>
```

```

<Resource>
  <Urn>urn:newsmml:iptc.org:20001006:IptcConfidence:1</Urn>
  <Url>http://www.iptc.org/NewsML/topicsets/iptc-confidence.xml</Url>
  <DefaultVocabularyFor Context="@Confidence"/>
</Resource>
<TopicUse Topic="#person1" Context="DescriptiveMetadata"/>
</Catalog>

<TopicSet FormalName="Person">
  <Topic Duid="person1">
    <TopicType FormalName="Person"
      Vocabulary="urn:newsmml:iptc.org:20001006:IptcTopicTypes:1" Scheme="IptcTopicTypes"/>
    <Description xml:lang="en">David Allen, Managing Director of IPTC</Description>
  </Topic>
</TopicSet>

```

Example 2:

```

<NewsML>
  <Catalog>
    <Resource>
      <Urn>...</Urn>
      <Url>...</Url>
      <DefaultVocabularyFor Context="//MediaType" Scheme="xyz"/>
    </Resource>
  ...
</NewsML>

```

[Catalog](#)の親であるNewsML要素から、すべての[MediaType](#)を探しはじめるには、次のように[Context](#)を設定する。

```
<DefaultVocabularyFor Context="//MediaType" Scheme="xyz"/>
```

NewsMLの定義では、`"/"`は[Catalog](#)要素の親から探す解釈される。`"/"`は[XPath](#)の定義としてはNewsML文書のルート要素からすべてを意味する。この場合でも、ルートは、[Catalog](#)要素の親であることに変わらない。

Example 3:

```

<NewsML>
  <NewsItem>
  ...
    <NewsComponent>
      <Catalog>
        <Resource>
          <Urn>...</Urn>
          <Url>...</Url>
          <DefaultVocabularyFor Context="//MediaType" Scheme="xyz"/>
        </Resource>
      ...
    </NewsML>

```

Example3の場合、[NewsComponent](#)から[MediaType](#)を探しはじめる。これはちょうど[Catalog](#)要素の親に相当する。しかし、次のように[Context](#)を設定すると

```
<DefaultVocabularyFor Context="//MediaType" Scheme="xyz"/>
```

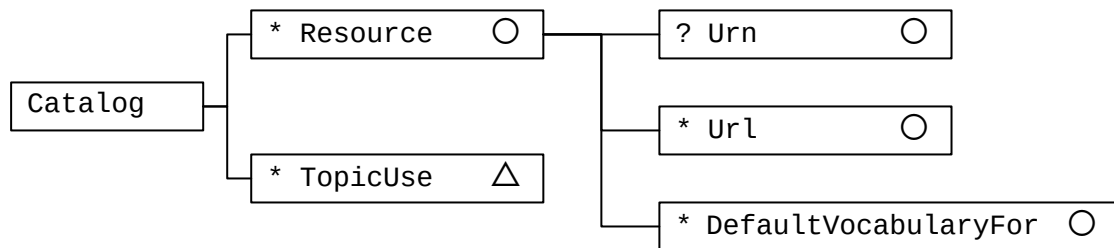
`"/"`はドキュメントの最初であるルートからを示すから、NewsML要素から探すことになる。

Example 4:

```
<NewsML>
  <NewsItem>
    <NewsComponent>
      <Catalog>
        <Resource>
          <Url>http://www.acmenews.com/vocabs/roles.xml</Url>
          <DefaultVocabularyFor Context="//Role"/>
        </Resource>
      </Catalog>
      <Role FormalName="alpha"/>
    </NewsComponent>
    <NewsComponent>
      <Role FormalName="beta"/>
    </NewsComponent>
  </NewsItem>
</NewsML>
```

この例では、XPath "//Role" は "alpha" と "beta" の <Role> 要素に合致する（なぜなら、両方とも document root の子孫だから）。しかし、NewsML では "beta" にこの "[DefaultVocabularyFor](#)" を適用することはできない。"Catalog" が先祖要素内に現れていないからだ。言いかえると、[XPath](#) の適用範囲は先祖ツリー内である。つまり、[XPath](#) がどのように書かれようともそれ以上広げることは許されていない。

[ツリー図]



[NSK 解説]

Catalog要素

Catalog要素は下位要素にResource要素とTopicUse要素を持つことができ、つぎの役割をもつ。

1. NewsML 文書に関連する外部ファイルなどのリソースを定義する。

(Resource要素下のUrl要素)

2. NewsML 文書内で使われる NewsML URN 等とリソースの対応付けを行う。

(Resource要素下のUrn要素)

3. リソースと、関連する NewsML 文書内の要素を結びつける宣言を行う。

(Resource要素下のDefaultVocabularyFor要素)

4. 特定の要素に対する Scheme属性と Vocabulary属性が省略されたときの省略値を宣言する。

(Resource要素下のDefaultVocabularyFor要素)

5. 実世界事情と呼ばれる Topicと、関連する NewsML 文書内の要素を結びつける宣言を行う。

(TopicUse要素)

上記の役割一覧内の NewsML URN とは、後述のNewsItem の章にある PublicIdentifier の値に示されるもので、NewsML 文書を唯一に表すことができる識別子である。これを用いて外部の NewsML 文書をポイントすることができる。

また、上記の役割一覧内の Topicとは、後述のTopicSet の章にある Topicのことであり、そこで説明されている 2 種類ある Topicの用途のうち、実世界事情と呼ばれる方である。

[Catalog](#)要素の記述方法

[Catalog](#)要素の記述方法は2つある。一つは、[Catalog](#)実体をサブツリーとして該当 NewsML 文書の中で直接定義する方法である。

Catalog の記述方法-1 (Catalog 実体を NewsML 文書内で直接定義する方法)

```
<NewsML>
  <Catalog>
    <Resource>
      <Url> ... </Url>
      <DefaultVocabularyFor ... />
    </Resource>
  </Catalog>
```

[Catalog](#)要素のもう一つの記述方法は、[Href](#)属性を用いた[Catalog](#)実体を参照するポインタを持つ方法である。この[Catalog](#)実体は Catalog ファイルと呼ばれる別ファイルに定義することが多い。[Href](#)の値は NewsML URN 以外に、http:, ftp: などを含む[URI](#)全般が設定可能である。相対パスは、"./" で始める。

Catalog の記述方法-2a (別ファイルにある Catalog 実体をポイントする方法)

```
<NewsML>
  <Catalog Href="./catalog/catalog.iptcmastercatalog.xml"/>
```

別途 URL で参照できる./catalog/catalog.iptcmastercatalog.xml に次の内容を用意しておく。

[Catalog ファイルの記述例]

```
<NewsML>
:
  <News Item>
    <NewsManagement>
      <NewsItemType FormalName="Catalog" Scheme="IptcNewsItemType"/>
      :
    <DataContent>
      <Catalog>
        <Resource>
          <Url> ... </Url>
          <DefaultVocabularyFor ... />
        </Resource>
      </Catalog>
    </DataContent>
  :
</NewsML>
```

Catalog の記述方法-2b (文書内にある Catalog 実体をポイントする方法-この方法はあまり使われない)

```
<Metadata>
  <Catalog Href="#catalog-duid-001"/>
```

その文書内にある Catalog 実体の例

```
  <Catalog Duid="catalog-duid-001">
    <Resource>
      <Url> ... </Url>
      <DefaultVocabularyFor ... />
    </Resource>
  </Catalog>
```

つぎのような[Href](#)属性を持ちながら、実体のサブツリーを持つ Catalog 記述はエラーである。

Catalog の記述誤り

```
<NewsML>

  <Catalog Href="...">
    <Resource>
      <Url> ... </Url>
      <DefaultVocabularyFor ... />
    </Resource>
  </Catalog>
```

[Catalog](#) 要素の記述位置と適用範囲 (スコープ)

[Catalog](#) 要素は、NewsML 文書中のいくつかの個所に記述できる。記述した個所によって、[Catalog](#) の適用される範囲が異なる。これを [Catalog](#) の適用範囲 (スコープ) と呼ぶ。[Catalog](#) のスコープは、その [Catalog](#) 要素を持つ親要素のサブツリー全体である。複数の [Catalog](#) のスコープに重なる場所では、下の図に表されたように内側のスコープから順に適用される。つまり内側のスコープが優先される。

Catalog の適用範囲 (スコープ)

```
<NewsML>
  <Catalog Href=" ./A.xml" />
  <NewsItem>
    <Catalog Href=" ./B.xml" />
    <NewsComponent>
      <Catalog Href=" ./C.xml" />
      <Metadata>
        <Catalog Href=" ./D.xml" />
        :
      </Metadata>
      <ContentItem>
        <Catalog Href=" ./E.xml" />
        :
      </ContentItem>
      <ContentItem>
        :
      </ContentItem>
```

A.xml の Catalog が適用される

B.xml, A.xml の順で Catalog が適用される

C.xml, B.xml, A.xml の順で Catalog が適用される

D.xml, C.xml, B.xml, A.xml の順で Catalog が適用される

E.xml, C.xml, B.xml, A.xml の順で Catalog が適用される

Resource要素

関連リソース実体の宣言と唯一の名称を対応づける機能を提供

[Resource](#)要素は、[Catalog](#)の下位要素であり、[Url](#)要素によってNewsML 文書に関連する外部ファイルなどのリソースを宣言する。また、[Urn](#)要素によって、文書内でNewsML URN を用いてリソースを指定しているときの、それに対応する実体を表すことができる。

Resource要素のUrlとUrnの記述例

```
<Resource>
  <Urn>urn:newsml:iptc.org:20001006:IptcConfidence:1</Urn>
  <Url>http://www.iptc.org/NewsML/topicsets/iptc-confidence.xml</Url>
  <Url>http://www.mycompany.com/NewsML/topicsets/iptc-confidence.xml</Url>
</Resource>
```

[Resource](#)要素下の[Urn](#)要素の値にはURNを設定する。URN (Uniform Resource Name) は、リソースをその実体の存在場所に依存せずに「名前」によって永続的に特定する識別子であり、urn:で始められる。urn:以降の記述形式は登録した団体によって決められている。urn:newsml:は、NewsML URN としてIPTCがIANAに登録したものである。<http://www.iana.org/assignments/urn-namespaces>を参照。

この[Urn](#)要素にはNewsML URN 以外のURNを指定することが可能であるが、外部リソースとしてNewsML 文書を指定することが多いため、通常は urn:newsml:で始まる。このNewsML URN は、NewsML 文書を世界で唯一に特定できる識別子であり、詳細は後述の[NewsItem の章](#)にある[PublicIdentifier](#)に記述されている。

[Url](#)要素は、リソース実体の場所が記述される。ここには、http:から始まるhttp URLや、ftp:から始まるftp URLが記述できる。[Url](#)要素を複数記述することによって、実体が複数の場所にあることを表すことができる。また、ここに#ではじまるDuidを記述することによって、内部に組み込んだリソースをポイントすることもできる。

URL に文書内の Duid を指定した例

```
<Catalog>
  <Resource>
    <Url>#LocalTopicSet</Url>
    <DefaultVocabularyFor Scheme="companycode" Context="@AssignedBy"/>
  </Resource>
</Catalog>
<TopicSet Duid="LocalTopicSet">
  <Topic Duid="company1">
    :
```


属性値に[Controlled Vocabulary](#)が使われることを宣言する機能を提供

[Resource](#)要素の下に[DefaultVocabularyFor](#)要素を記述することによって、特定要素の属性値に[Controlled Vocabulary](#)が使われることを宣言できる。[Controlled Vocabulary](#)とは、「候補値の一覧」と考えればよく、NewsML 文書内に設定すべき値を（例えば属性ごとのように）特定単位ごとに、その設定値の候補になるものを集めた値の集合（ボキャブラリ、語彙）のことである（[次章 TopicSetの章](#)、および仕様書用語集参照）。

[IPTC](#)が提供している[Controlled Vocabulary](#)のファイル名とその概要

(http://www.newsml.org/pages/spec_main.php の「a full [NewsML 1.x package](#)」より 2004/1/17 現在)

Controlled Vocabularyの名称	概要（何の候補値かを表している）
topicset.iptc-audiocoder.xml	オーディオ・コーダー種別
topicset.iptc-characteristicsproperty.xml	Characteristics/PropertyのFormalName用
topicset.iptc-colorspace.xml	カラースペース
topicset.iptc-confidence.xml	信頼度
topicset.iptc-encoding.xml	エンコーディング種別
topicset.iptc-format.xml	Format種別
topicset.iptc-genre.xml	記述形式のジャンル
topicset.iptc-howpresent.xml	適用理由
topicset.iptc-importance.xml	メタデータの重要度
topicset.iptc-labeltype.xml	Label種別
topicset.iptc-mediatype.xml	メディア種別
topicset.iptc-mimetype.xml	Mime種別
topicset.iptc-newsitemtype.xml	NewsItem種別
topicset.iptc-notation.xml	Notation種別
topicset.iptc-ofinterestto.xml	関心のある人々
topicset.iptc-priority.xml	配信の優先度
topicset.iptc-property.xml	Property用
topicset.iptc-provider.xml	配信社
topicset.iptc-relevance.xml	適切度
topicset.iptc-role.xml	役割
topicset.iptc-status.xml	NewsItemの状態
topicset.iptc-subjectcode.xml	サブジェクト・コード
topicset.iptc-subjectqualifier.xml	サブジェクト・コード補足
topicset.iptc-topictype.xml	Topic種別
topicset.iptc-urgency.xml	NewsItemの編集上の重要性を伴う緊急度
topicset.iptc-videocoder.xml	ビデオ・コーダー
topicset.iso-country.xml	国名
topicset.iso-currency.xml	通貨
topicset.iso-language.xml	言語
topicset.naics-classofindustry.xml	NAICS業界種別
topicset.nasdaq-company.xml	NASDAQ会社コード
topicset.uno-regions.xml	世界の地域

Controlled Vocabularyの宣言方法

Resource要素内に、DefaultVocabularyFor要素を記述することによって、この Resource 要素の Url もしくは Urnが Controlled Vocabularyとして使われることを表す。

Controlled Vocabularyを適用する属性の指定方法

DefaultVocabularyFor下位要素の Context属性には、Controlled Vocabularyを使う特定要素の属性を XPath表記で記述する。DefaultVocabularyFor要素の Scheme属性は、候補値を明確に示すために記述する。NewsML 文書内のすべての属性に Controlled Vocabularyを使うことができる。要素値には適用することができない。

Controlled Vocabulary を適用する特定要素属性の指定例

```
<Resource>
  <Urn>urn:newsml:iptc.org:20001006:IptcConfidence:1</Urn>
  <DefaultVocabularyFor Scheme="IptcConfidence" Context="//Subject/@Confidence"/>
</Resource>
```

XPath表記は、カレント・ポイントを Catalog要素の親要素として表記する。XPath表記の仕方によっては、“/”から始めることによって文書のルート要素から指定することも可能だが、Catalogのスコープ内だけが有効である。

XPath と Catalog スコープの関係

```
<NewsItem>
  <NewsComponent>
    <Catalog>
      <Resource>
        <Urn>urn:newsml:iptc.org:20001006:IptcConfidence:1</Urn>
        <DefaultVocabularyFor Scheme="IptcConfidence" Context="//Subject/@Confidence"/>
      </Resource>
    </Catalog>
    :
    <Subject Confidence="..."/>.....ここには適用される
    :
  </NewsComponent>

  <NewsComponent>
    :
    <Subject Confidence="..."/>.....Catalogスコープからはずれるため、適用されない
    :
  </NewsComponent>
</NewsItem>
```

DefaultVocabularyFor要素のContext属性に記述するXPath表記の特例と複雑な例

Context属性に記述するXPath表記には、さまざまな特例がある。

1. “/”, “.”, “@”から始まらないときは、“./” (カレント・ポイントからのすべてのサブツリーにある要素) が省略されていることを表す。

例: “Property/@FormalName”? “./Property/@FormalName”

(カレント・ポイントからのすべてのProperty要素のFormalName属性)

2. 最後の要素の後ろに“/@”から始まる属性名が記述されていないときは、“/@FormalName”が省略されていることを表す。これは頻繁に使われる形式である。

例: “./Status”? “./Status/@FormalName”

(カレント・ポイントからのすべてのサブツリーのStatus要素のFormalName属性)

3. “@”から始まるときは、“./*/” (カレント・ポイントからのすべてのサブツリーのすべての要素の) が省略されていることを表す。

例: “@Importance”? “./*/@Importance”

(カレント・ポイントからのすべてのサブツリーのすべての要素のImportance属性)

つぎは特例ではないが、要素名が同じでも属性値が特定の場合だけに適用する例を示す。

例: “Property[@FormalName='ColorSpace']/@Value”

(カレント・ポイントからのすべてのサブツリーのすべてのProperty要素で、FormalName属性値が“ColorSpace”と等しいときのValue属性)

複数のContext属性のXPath表記に一致する属性

複数階層の位置にCatalog要素を記述できるため、DefaultVocabularyFor要素のContext属性のXPath表記を評価すると、同一要素の同一属性を示す場合がある。この場合、Catalogのスコープによる優先順位に伴ってどのDefaultVocabularyFor要素を持つResourceを適用するかを決定する。

DefaultVocabularyFor要素のContext属性のXPath表記を評価した結果、一つのCatalog内から同一要素の同一属性を示す場合は処理系によって解釈が異なる可能性があり、このような表現をしないことが望ましい。

DefaultVocabularyFor要素が複数連続して記述されている場合

一つのResource要素の中に複数のDefaultVocabularyFor要素があり、Context属性の値も同じ場合がある。このような場合、Scheme属性値が複数提供される。どちらのSchemeを先に探すかは、明確に規定されていないため、探す順序によって結果が異なるControlled Vocabularyを作成することは推奨しない。

一つの属性に対して、複数のScheme属性を指定した例

```
<Catalog>
  <Resource>
    <Url>topicset.iptc-format.xml</Url>
    <DefaultVocabularyFor Scheme="IptcFormat" Context="//Format"/>
    <DefaultVocabularyFor Scheme="NskFormat" Context="//Format"/>
  </Resource>
</Catalog>
```

上記の場合、Format要素のFormalName属性にControlled Vocabularyが適用されるが、2種類のScheme属性値が提供される。たとえばControlled Vocabularyから、先にSchemeが"IptcFormat"で、一致するFormalNameを探し、見つからなければSchemeが"NskFormat"で一致するFormalNameを探す。

同じようにDefaultVocabularyFor要素が連続していても、Context属性が異なる場合もある。この場合は、複数の要素に同じControlled Vocabularyが使われることを意図している。このような場合は、Scheme属性が異なり、Contextに記述されたXPath表記から求められる属性値ごとにSchemeを変えることによってControlled Vocabularyの候補を絞り込むような工夫がされていることがある。

一つのリソースを複数のContextに利用する例

```
<TopicSet Duid="LocalTopicSet" Scheme="IptcTopicType" FormalName="LocalVocabulary">
  <Topic Duid="BooleanTrue">
    <TopicType Scheme="IptcTopicType" FormalName="Property"/>
    <FormalName Scheme="Boolean">True</FormalName>
  </Topic>
  <Topic Duid="BooleanFalse">
    <TopicType Scheme="IptcProperty" FormalName="Property"/>
    <FormalName Scheme="Boolean">False</FormalName>
  </Topic>
  <Topic Duid="DayMonday">
    <TopicType Scheme="IptcProperty" FormalName="Property"/>
    <FormalName Scheme="Day">Monday</FormalName>
  </Topic>
  <Topic Duid="DayTuesday">
    <TopicType Scheme="IptcProperty" FormalName="Property"/>
    <FormalName Scheme="Day">Tuesday</FormalName>
  </Topic>
</TopicSet>
...
<Catalog>
  <Resource>
    <Url>#LocalTopicSet</Url>
    <DefaultVocabularyFor Scheme="Boolean"
      Context="Property[@FormalName='Print']/@Value"/>
    <DefaultVocabularyFor Scheme="Day"
      Context="Property[@FormalName='PrintDay']/@Value"/>
  </Resource>
</Catalog>
...
<Property FormalName="Print" Value="True"/>
<Property FormalName="PrintDay" Value="Monday"/>
```

Controlled Vocabularyの候補値の見つけ方

[DefaultVocabularyFor](#) 要素によって、特定要素の属性値が [Controlled Vocabulary](#) によって制御されることになっても、さらに [Scheme](#) 属性で候補が絞られると考える必要がある。この候補に一致する条件はつぎのとおりである。

[DefaultVocabularyFor](#) 要素の [Context](#) 属性値の [XPath](#) 表記が示す対象の属性値が [TopicSet](#) の [FormalName](#) 要素値と一致し、[DefaultVocabularyFor](#) 要素の [Scheme](#) 属性値が [TopicSet](#) の [Scheme](#) 属性値と一致する [Topic](#) が存在する必要がある。

候補値の見つけ方例

下記 AssignedBy 属性に入れることができる値の候補を Description で表すと、Takahiro と Manabu である。

```
<Resource>
  <Urn>#assign-Duid-001</Urn>
  <DefaultVocabularyFor Scheme="Japan" Context="@AssignedBy"/>
</Resource>
:
<SubjectCode AssignedBy="...">

<TopicSet Duid="assign-Duid-001">
  <Topic>
    <TopicType FormalName="Person"/>
    <FormalName Scheme="Japan">1</FormalName>
    <Description>Takahiro</FormalName>
  </Topic>
  <Topic>
    <TopicType FormalName="Person"/>
    <FormalName Scheme="Japan">2</FormalName>
    <Description>Manabu</Descriptiton>
  </Topic>
  <Topic>
    <TopicType FormalName="Person"/>
    <FormalName Scheme="France">1</FormalName>
    <Description>Laurant</Description>
  </Topic>
</TopicSet>
```

[Scheme](#)属性値と[Controlled Vocabulary](#)の置き換え（上書き、オーバーライド）

[Resource](#)要素によって与えられた[Scheme](#)属性値と[Controlled Vocabulary](#)は、実際に出現した要素と属性の指定方法によっては、置き換えることができる。これを値の上書き、あるいは、値のオーバーライド（override）と呼ぶ。そして、[DefaultVocabularyFor](#)要素によって与えられた[Scheme](#)属性値のことを省略時 [Scheme](#)と呼び、[DefaultVocabularyFor](#)要素を持つ[Resource](#)の[Urn](#)または[Url](#)を省略時 [Vocabulary](#)と呼ぶ。なお、値のオーバーライドはつぎの2つの属性にしか適用できない。

1. すべての要素の[FormalName](#)属性

[FormalName](#)属性に対する省略時 [Scheme](#)のオーバーライドは[FormalName](#)属性を持つ要素に、[Scheme](#)属性を記述することによって表現される。

例：<NewsService FormalName="abc" Scheme="MyService"/>

（上記の FormalName="abc"には、省略時 Scheme は"MyService"でオーバーライドされ、省略時 [Vocabulary](#)が適用される。）

[FormalName](#)属性の省略時 [Vocabulary](#)に対するオーバーライドは[FormalName](#)属性を持つ要素に、[Vocabulary](#)属性を記述することによって表現される。このときに[Scheme](#)属性が省略されている場合でも、省略時 [Scheme](#)が Schemeの値が設定されていないものとして、オーバーライドされることに注意が必要である。

例：<NewsService FormalName="xyz" Vocabulary="/news.xml"/>

（上記の FormalName="xyz"には、省略時 Scheme は値が設定されていないものとしてオーバーライドされ、省略時 [Vocabulary](#)は"/news.xml"でオーバーライドされる。）

2. [Property](#)の[Value](#)属性

[Value](#)属性に対する省略時 [Scheme](#)のオーバーライドは[Value](#)属性を持つ要素に、AllowedScheme 属性を記述することによって表現される。AllowedScheme属性は NewsML1.1 で追加された。

例：<Property FormalName="Color" Value="red" AllowedScheme="Pantone"/>

（上記の Value="red"は、省略時 Scheme は"Pantone"でオーバーライドされ、省略時 [Vocabulary](#)が適用される。FormalName="Color"に対しては[Property](#)用の省略時 Scheme と省略時 [Vocabulary](#)が適用される。）

[Value](#)属性の省略時 [Vocabulary](#)に対するオーバーライドは[Value](#)属性を持つ要素に、[AllowedValues](#)属性を記述することによって表現される。このときに AllowedScheme 属性が省略されている場合でも、省略時 Scheme が AllowedScheme の値が設定されていないものとして、オーバーライドされることに注意が必要である。

例：<Property FormalName="Color" Value="red" AllowedValues="/rgb.xml"/>

（上記の Value="red"は、省略時 Schemeは値が設定されていないものにオーバーライドされ、省略時 [Vocabulary](#)は"/rgb.xml"でオーバーライドされる。FormalName="Color"に対しては[Property](#)用の省略時 Scheme と省略時 [Vocabulary](#)が適用される。）

ボキャブラリの決定アルゴリズム (DTDのVocabulary属性に関するコメントより)

NewsML 内の要素(のFormalName)や属性のcontrolled_vocabularyは次のアルゴリズムで探し当てられる (その要素や属性にVocabulary属性指定がない場合のボキャブラリの決定方法)

1. その要素から親をたどる。
2. その要素の親要素が直下の下位要素としてCatalog要素を持っているならば、そのCatalog要素が次の条件に合致するResource要素をもつかどうかを判断する。
「DefaultVocabularyFor/@Context属性値 (XPath 値) がその属性に合致するか」
合致すればそのResource要素のUrn要素またはUrl要素の値が該当するcontrolled_vocabularyである。
3. この条件に満ちたResource要素がない場合、さらに次の親要素を辿り、見つけたCatalog要素が上記条件に合致するResource要素を持つかどうかを判断する。
4. これを、ボキャブラリが見つかるまで繰り返す。最上位のCatalog(NewsML/Catalog)まで遡ってもボキャブラリが見つからなければFormalName属性に限ってはエラーである。その他の場合は自由形式を表すので、エラーにならない (NewsML v1.2 では、FormalName 属性においても、エラーとしないように変更された)。

つぎに、Catalog の使用例を示す

例 1 : NewsML/Catalog にのみ Catalog を宣言した場合。

```
<NewsML>
  <Catalog>
    <Resource> . . . . . ◆ 1
    <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.nsk-party:1</Urn>
    <DefaultVocabularyFor Scheme="NskParty" Context="//Party/@FormalName"/>
  </Resource>
</Catalog>

<NewsEnvelope>
  <SentFrom>
    <Party FormalName="KYODO"/>
    ?この Party/@FormalName 値"KYODO"は、上記 Resource ◆ 1で示される
      controlled vocabulary に定義されている。
  </SentFrom>
</NewsEnvelope>

<NewsItem>
  <NewsComponent>
    <AdministrativeMetadata>
      <Provider>
        <Party FormalName="KYODO"/>
        ?この Party/@FormalName 値にも同じく、
          ◆ 1 の controlled vocabulary が適用される。
      </Provider>
      <Creator>
        <Party FormalName="Reuter"/>
        ?この Party/@FormalName 値にも同じく、
          ◆ 1 の controlled vocabulary が適用される。
      </Creator>
    </AdministrativeMetadata>
  </NewsComponent>
</NewsItem>
</NewsML>
```

例 2 : NewsML/Catalog と AdministrativeMetadata/Catalog に Catalog が配置され、両方に Context="Party" の Resource が宣言されている場合。

```
<NewsML>
  <Catalog>
    <Resource> . . . . . ◆ 1
    <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.nsk-party:1</Urn>
    <DefaultVocabularyFor Scheme="NskParty" Context="//Party/@FormalName"/>
  </Resource>
</Catalog>
<NewsEnvelope>
  <SentFrom>
    <Party FormalName="KYODO"/>
    ?この Party/@FormalName 値"KYODO"は上記 Resource ◆ 1 で示される
    controlled vocabulary に定義されている。
  </SentFrom>
</NewsEnvelope>
<NewsItem>
  <NewsComponent>
    <AdministrativeMetadata>
      <Catalog>
        <Resource> . . . . . ◆ 2
        <Urn>urn:newsml:MyDomain:20010516:topicset.my-party</Urn>
        <DefaultVocabularyFor Scheme="MyParty" Context="//Party/@FormalName "/>
      </Resource>
    </Catalog>
    <Creator>
      <Party FormalName="Ken Domon"/>
      ?この値"Ken Domon"は、◆ 2 の controlled vocabulary に定義されている。
    </Creator>
  </AdministrativeMetadata>
</NewsComponent>
</NewsItem>
</NewsML>
```

[注意点]

IPTCNewsML1.05 以前は、[Catalog](#)の[Href](#)属性に、NewsML URN と、http URL が利用可能であったが、IPTCNewsML1.05 からは、NewsML URN と、[URI](#)となった。つまり、http:以外に、ftp:なども含まれるようになった。しかし、NSKNewsML レベル 1.2 では、レベル 1 との互換性を考え、NewsML URN と http URL のみを使用可能とする。Resourceの子要素の URI についても同様とする。

[NskNewsML:1 記述形式]

Catalog 要素は下記の複数の要素の下位要素であり、NewsML:1 での使用許可は次の通り。

場所	使用許可	備考
NewsML/Catalog	◎	
NewsItem/Catalog	○	
TopicSet/Catalog	△	TopicSet 要素は、次の複数の要素の下位要素である。 NewsML 、NewsItem、NewsComponent 上記すべてについて TopicSet/Catalog は△。
TopicSet/Topic/Catalog	△	Topic 要素は、TopicSet 要素の下位要素であり、上欄に示すとおり複数の要素内に現れるがいずれも△。
NewsComponent/Catalog	○	
AdministrativeMetadata/Catalog	○	
RightsMetadata/Catalog	○	
DescriptiveMetadata/Catalog	○	
Metadata/Catalog	○	
ContentItem/Catalog	○	
DataContent/Catalog	○	Catalog/@Href で参照されるファイルを作る場合には、ここを使用すること。

Catalog 要素の下位要素

- Catalog/Resource 要素……………○
- Catalog/Resource/Urn 要素……………○
- Catalog/Resource/Url 要素……………○
- Catalog/Resource/DefaultVocabularyFor 要素……………○
- Catalog/Resource/DefaultVocabularyFor/@Context 属性……○
- Catalog/Resource/DefaultVocabularyFor/@Scheme 属性……○（注1）

注1．Scheme 属性のある Topic を指定する場合に、FormalName 属性だけで良いとしていたが、IPTC の通信社ガイドラインに基づき Scheme 属性もセットすることに変更する。

Scheme をつける場合は Catalog/Resource/DefaultVocabularyFor の@Scheme 属性か、各要素の Scheme 属性のどちらを使用しても良い。両方につけることも可である。

TopicSet

TopicSetsもNewsMLの各階層に置くことができる。TopicSetsを含むことができる要素は、NewsML、NewsItem、NewsComponentの各要素である。TopicSet要素がこれらの要素内にある場合は、Local Vocabulariesと呼ばれる。TopicSetはVocabulary属性を使用することで指定される。

Vocabularyには、そのFormalNameの意味を解析するために使えるcontrolled vocabularyであるTopicSetへのポインターを与える。Vocabulary属性の値はhttp URLか、(外部Vocabulariesの)NewsML URNか、#文字に要素の存在する文書中のTopicSet (Local Vocabularyのこと)のDuidを続けるかのいずれかである。Vocabularyポインターは特定のTopicSetを識別できるだけなので、TopicSetの所在は継承されない。けれども、VocabularyポインターはCatalogによって指し示されたデフォルト・ボキャブラリーよりも優先して使われる。

(Topicの) FormalName要素は、特定のネーミング・スキームに属することを示すためにScheme 属性を持つことができる。同じScheme属性を持ち、FormalNameが同じであるTopicが同じTopicSetに存在すればエラーとなる。Scheme属性が存在すれば、controlled vocabularyにある複数のネーミング・スキームのうち、どれがこのFormalNameを管理するものであるかを区別するために利用できる。controlled vocabularyから一致するTopicを探すには、FormalNameとSchemeがともに一致しなければならない。当該の要素にScheme属性がなければ、当該のFormalNameを持ち、Schemeを持たないローカルのボキャブラリー内のアイテムと一致しなければならない。

当該の要素にScheme属性があれば、ローカルのcontrolled vocabulary中のFormalNameとSchemeがともに一致しなければならない。外部VocabularyがCatalogからポイントされると、そのCatalogの関連Resourceの下位要素であるDefaultVocabularyForにおけるScheme属性値によって適切なSchemeが表示される。カレント要素内にFormalName属性値を持つScheme属性が存在しなければ、CatalogのDefaultVocabularyFor エントリーのSchemeの記述がその要素にとってデフォルトとなる。

問題を避けるために、Scheme属性が常に使われ、積極的にTopicSetにおける1つのTopicのFormalNameとSchemeの値をともに用いて、必ず一致させるようにすることが望ましい。

Local Topicは、TopicSet Ref要素を使用して外部のデフォルトcontrolled vocabulariesに追加されることもある。TopicSet Refで指定するのはLocal TopicにマージされるTopicSetへのポインターである。TopicSet 属性は関連するTopicSetへのポインターである。http URLか、NewsML URNか、#文字にこの文書中のTopicSetのDuidを続ける形のfragment identifierのいずれかの値をとることができる。TopicSet内に下位要素のTopicSetRefがあれば、参照されたTopicSetのTopicsがすべて当該のローカルなTopicSet内で参照によって内包されるという効果がある。このマージによって、同じTopicSetの孫要素である2つのFormalName要素が、同じ内容と同じScheme 属性値を持つ場合は、これら2つの要素が実際には同じTopicであり、Topicsはマージされと考えられる。Topicsのマージはシステムによって物理的に行う必要はないが、意味はマージが実際に行われたことと同じである。2つのTopicのマージは、両方のTopicが持つ下位要素をすべて含む1つのTopicを作成し、重複を避けることに行われる。この方法により、デフォルトの外部TopicSetにあるTopicsにSchemeを付け加えることができる。

例) [Catalog](#) で Scheme を指定

```
<NewsML>
  <Catalog>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-newsitemtype-ja:4</Urn>
      <Url>../topicsets/topicset.iptc-newsitemtype-ja.xml</Url>
      <DefaultVocabularyFor Scheme="IptcNewsItemType"
        Context="//NewsItemType/@FormalName"/>
    </Resource>
  </Catalog>
  <NewsItem>
    .....
  <NewsManagement>
```

```
<NewsItemType FormalName="TopicSet"/>
.....
```

例) 要素で Scheme を宣言

```
<NewsML>
  <Catalog>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-newsitemtype-ja:4</Urn>
      <Url>../topicsets/topicset.iptc-newsitemtype-ja.xml</Url>
      <DefaultVocabularyFor Context="//NewsItemType/@FormalName"/>
    </Resource>
  </Catalog>
  <NewsItem>
    .....
    <NewsManagement>
      <NewsItemType FormalName="TopicSet" Scheme="IptcNewsItemType" />
    .....
  </NewsItem>
</NewsML>
```

◆ NskNewsML:1 リソースセット

NskNewsML:1 では、利用を指定または推奨する Resource のリストを用意している。これを「NskNewsML リソースセット」と呼ぶ。NskNewsML:1 では利用者は、このリソースセットの中にある Resource をベースに Catalog を構成する。

Catalog は Href 属性によって外部ファイルとして参照してもよい。

外部ファイルとする場合、DataContent 要素内に Catalog 要素を配置し、NewsML ドキュメントとする。

下記の URN で示される IPTC の MasterCatalog を参照。

urn:newsml:iptc.org:20001006:catalog.IptcMasterCatalog:20
IPTC のサイト http://www.newsml.org/pages/spec_main.php の「a full [NewsML 1.x package](#)」より
(2004/1/17 現在)

NskNewsML リソースセットに含まれる controlled vocabulary については、5.3 TopicSet に記述する。

----- NskNewsML:1 リソースセット -----

以下のリソースの一部を必要に応じて適宜 Catalog 内に配置し、NewsML ドキュメントを構成する。このリソースセットの Resource 要素に Url 要素を入れることもできる。さらにリソースが必要な場合は下記に準じた方法で追加する。

注：このリソースセットは 2005 年 7 月 25 日現在のサンプル（2005 年 6 月 1 日版）です。最新の TopicSet のリビジョンに関しては日本新聞協会のウェブサイト（<http://www.newsml.jp/>）を参照してください。

```
<Resource>
  <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.iptc-format-nsk:10
</Urn>
  <!-- 下記 Scheme は NskFormats 等になることもある -->
  <DefaultVocabularyFor Scheme="IptcFormat" Context="//Format/@FormalName"/>
</Resource>
<Resource>
  <Urn>urn:newsml:iptc.org:20001006:topicset.iso-language-ja:3</Urn>
  <DefaultVocabularyFor Scheme="ISO639" Context="//Language/@FormalName"/>
</Resource>
</Resource>
```

```

        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-mediatype-ja:4</Urn>
        <DefaultVocabularyFor Scheme="IptcMediaTypes" Context="//MediaType/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.iptc-metadatatype-nsk:2</Urn>
        <!-- 下記 Scheme は NskMetadataTypes 等になることもある -->
        <DefaultVocabularyFor Scheme="IptcMetadataTypes" Context="//MetadataType/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-mimetype-ja:4</Urn>
        <DefaultVocabularyFor Scheme="IptcMimeTypes" Context="//MimeType/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-newsitemtype-ja:4</Urn>
        <DefaultVocabularyFor Scheme="IptcNewsItemType" Context="//NewsItemType/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.nsk-newslinetype:2</Urn>
        <DefaultVocabularyFor Scheme="NskNewsLineType" Context="//NewsLineType/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.nsk-newsproduct:2</Urn>
        <DefaultVocabularyFor Scheme="NskNewsProduct" Context="//NewsProduct/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.nsk-jpnareacode:4</Urn>
        <DefaultVocabularyFor Scheme="Code"
            Context="//Property[@FormalName='NskJpnAreaCode']/@Value"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iso-country-ja:4</Urn>
        <DefaultVocabularyFor Scheme="ISO3166-alpha3"
            Context="//Property[@FormalName='NskCountry']/@Value"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.nsk-party:6</Urn>
        <!-- 下記 Scheme は NskTiffServiceId 等になることもある -->
        <DefaultVocabularyFor Scheme="NskParty" Context="//Party/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-priority-ja:2</Urn>
        <DefaultVocabularyFor Scheme="IptcPriority" Context="//Priority/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.iptc-property-nsk:4</Urn>
        <DefaultVocabularyFor Scheme="IptcProperty" Context="//Property/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-status-ja:2</Urn>
        <DefaultVocabularyFor Scheme="IptcStatus" Context="//Status/@FormalName"/>
        <DefaultVocabularyFor Scheme="IptcStatus" Context="//FutureStatus/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-subjectcode-ja:13</Urn>
        <DefaultVocabularyFor Scheme="IptcSubjectCodes" Context="//Subject/@FormalName"/>
        <DefaultVocabularyFor Scheme="IptcSubjectCodes"
            Context="//SubjectMatter/@FormalName"/>
        <DefaultVocabularyFor Scheme="IptcSubjectCodes"
            Context="//SubjectDetail/@FormalName"/>
    </Resource>
    <Resource>
        <Urn>urn:newsml:iptc.org:20001006:topicset.iptc-subjectqualifier-ja:10</Urn>
        <DefaultVocabularyFor Scheme="IptcSubjectQualifiers"
            Context="//SubjectQualifier/@FormalName"/>

```

```
</Resource>
<Resource>
  <Urn>urn:newsml:pressnet.or.jp:20010516:topicset.iptc-topictype-nsk:5</Urn>
  <!-- 下記 Scheme は NskTopicType 等になることもある -->
  <DefaultVocabularyFor Scheme="IptcTopicType" Context="//TopicType/@FormalName"/>
  <!-- 下記 Scheme は NskTopicType 等になることもある -->
  <DefaultVocabularyFor Scheme="IptcTopicType" Context="//TopicSet/@FormalName"/>
</Resource>
```

◆ボキャブラリの種類

NskNewsML:1 では、ボキャブラリを派生の仕方によって次の種類に分けた。

	ボキャブラリの種類					
	IPTC ボキャブラリ	IPTC-ja ボキャブラリ	IPTC-NSK 拡張 ボキャブラリ	NSK オリジナル ボキャブラリ	ユーザー拡張 ボキャブラリ	ユーザー ボキャブラリ
説明	IPTC の用意するボキャブラリ。	IPTC ボキャブラリの各 Topic に、日本語訳 Description を追加したもの。	IPTC-ja ボキャブラリに、NSK が Topic を追加したボキャブラリ。	NSK が独自に作成したボキャブラリ。	IPTC や NSK の用意するボキャブラリに、ユーザーが Topic を追加したり、Description を追加するなどしたボキャブラリ。	ユーザーが独自に作ったボキャブラリ。すでにある TopicSet の一部を組み込んだボキャブラリもこれに該当する。
ネーミング 説明		オリジナル名の末尾に "-ja" を付ける	オリジナル名の末尾に "-nsk" をつける	IPTC のネーミングに倣う	末尾に識別子をつける	他のボキャブラリと混同しないように命名する
ネーミング サンプル	topicset.iptc-format	topicset.iptc-subjectcode-ja、 topicset.iso-country-ja	topicset.iptc-format-nsk	topicset.nsk-NewsProduct	topicset.nsk-party-xxx	topicset.xxx

◆NSK リソース利用方法一覧

以下の表は、Context をキーとして、1 行が Resource ひとつに対応している。

Context の主要部分	ボキャブラリタイプ				
	IPTC-ja ボキャブラリ	IPTC-NSK 拡張 ボキャブラリ	NSK オリジナル ボキャブラリ	ユーザー拡張 ボキャブラリ	ユーザー ボキャブラリ
Format/@FormalName	使用不可	使用可(推奨) topicset.iptc-format-nsk	なし	使用可	使用不可 (ユニーク性を維持するため)
Language/@FormalName	使用可 topicset.iso-language-ja	なし	なし	使用不可	使用不可
MediaType/@FormalName	使用可 topicset.iptc-mediatype-ja	なし	なし	使用不可	使用不可
MetadataType/@FormalName	使用不可	使用可 topicset.iptc-metadatatype-nsk	なし	使用可	使用可
MimeType/@FormalName	使用可 topicset.iptc-mimetype-ja	なし	なし	使用不可	使用不可
NewsItemType/@FormalName	使用可 topicset.iptc-newsitemtype-ja	なし	なし	使用不可	使用不可
NewsLineType/@FormalName	なし	なし	使用可 topicset.nsk-newslinetype	使用可	使用不可
NewsProduct/@FormalName	なし	なし	使用可 topicset.nsk-newsproduct	使用不可	使用不可
Property[@FormalName= NskJpnAreaCode 1]/@Value	なし	なし	使用可 topicset.nsk-jpnareacode	使用不可	使用不可
Property[@FormalName= NskCountry 1]/@Value	使用可 topicset.iso-country-ja	なし	なし	使用不可	使用不可
Party/@FormalName	なし	なし	使用可(推奨) topicset.nsk-party	使用可	使用可 (ユーザーボキャブラリの使用の可能性が高いため)
Priority/@FormalName	使用可 topicset.iptc-priority-ja	なし	なし	使用不可	使用不可
Property/@FormalName	使用不可	使用可 topicset.iptc-property-nsk	なし	使用可	使用可
Status/@FormalName Future Status/@FormalName	使用可 topicset.iptc-status-ja	なし	なし	使用不可	使用不可

Context の主要部分	ボキャブラリタイプ				
	IPTC-ja ボキャブラリ	IPTC-NSK 拡張 ボキャブラリ	NSK オリジナル ボキャブラリ	ユーザー拡張 ボキャブラリ	ユーザー ボキャブラリ
Subject/@FormalName SubjectMatter/@FormalName SubjectDetail/@FormalName	使用可 topicset.iptc-subjectcode-ja	なし	なし	使用不可	使用不可
SubjectQualifier/@FormalName	使用可 topicset.iptc-subjectqualifier-ja	なし	なし	使用不可	使用不可
TopicType/@FormalName TopicSet/@FormalName	使用不可	使用可 topicset.iptc-topictype-nsk	なし	使用可	使用不可 (ユニーク性を維持するため)

5.3 TopicSets

[TopicSet](#)は[Topic](#)要素を含む。これら（[Topic](#)要素）は、実世界事情（トピック）への参照となる。これらは人、場所、会社、あるいはある種の重要性をもつその他のものであるかもしれない。そして、NewsML文書にあるニュース・コンテンツかメタデータ内で参照されるか、さもなければ関連しているかのどちらかである。

ひとつの[Topic](#)は、1つ以上の[FormalName](#)下位要素、1つ以上の[Description](#)下位要素の両方、あるいはどちらかを持ち得る。[Description](#)は、それがどの個別のものであるかについて識別するように意図されるものである。[FormalName](#)要素は、特定の[naming scheme](#)に属することを示すために、[Scheme](#)属性を持つことができる。同じ[Scheme](#)属性を伴う同じ[FormalName](#)を持つ2つの[Topic](#)が同じ[TopicSet](#)に存在するのは誤りである。従って[controlled vocabulary](#)として[TopicSet](#)を使うことは可能であり、すべての与えられた formal name（形式名）の意味を確かめることができるのである。

[Topic](#)要素は、[Details](#)属性を持つことができる。[Details](#)属性は[URL](#)や[URN](#)の形で、そのトピックに関する追加情報へのポイントとなる。[Topic](#)要素はまた、そのトピックの特性のための値を提供する1つか複数の[Property](#)下位要素を持つことができる。[Topic](#)と[TopicSet](#)は追加的に、自然言語で非形式的な追加情報を与える[Comment](#)を持つことができる。

追加の[Topic](#)は、[TopicSetRef](#)下位要素を使用することにより、ある[TopicSet](#)に参照として含まれることができる。[TopicSetRef](#)要素の[TopicSet](#)属性は、現在の[TopicSet](#)の中へ参照によって含まれる[Topic](#)をもつ、[TopicSet](#)へのポイントである。このポイントは、内部あるいは外部の[TopicSet](#)を識別する http URL または NewsML URN か、あるいは#記号の後に現在の文書中にある[TopicSet](#)の [Duid](#)属性値を続けたものからなる[fragment identifier](#)のいずれかある。

もし、参照によって含まれる [Topic](#) の1つが、すでにその[TopicSet](#)に含まれるある [Topic](#)と同じ [FormalName](#)と [Scheme](#)を持っていれば、それら両方が同じ実世界のものを参照するということを意味する。したがって、これら2つの [Topic](#) 要素は、マージされるとみなされる。[Topic](#)のマージは、システムによって物理的に実行される必要はない。しかし、データの意味は、まるでマージが実際に実行された場合と全く同じようになる。

すべての[Topic](#)は、その[Topic](#)がどのようなタイプであるかを表す1つまたはそれ以上の[TopicType](#)下位要素を持つ。トピックのタイプは、[TopicType](#)要素の[FormalName](#)属性で命名される。[TopicType](#)要素の[Vocabulary](#)属性は、その[FormalName](#)の意味を定義する[controlled vocabulary](#)へのポイントである。[Scheme](#)属性は、もし存在するならば、ボキャブラリ内のどの[naming scheme](#)がこの formal name（形式名）に当てはまるかを識別する。

```
<!ENTITY % formalname " FormalName CDATA #REQUIRED
                        Vocabulary CDATA #IMPLIED
                        Scheme CDATA #IMPLIED" >

<!ELEMENT TopicSet (Comment* , Catalog? , TopicSetRef* , Topic*)>
<!ATTLIST TopicSet %localid;
                %formalname; >

<!ELEMENT TopicSetRef (Comment*)>
<!ATTLIST TopicSetRef %localid;
                TopicSet CDATA #IMPLIED >

<!ELEMENT Topic (Comment* , Catalog? , TopicType+ , FormalName* , Description* , Property*)>
<!ATTLIST Topic %localid;
                Details CDATA #IMPLIED >

<!ELEMENT TopicType EMPTY >
<!ATTLIST TopicType %localid;
                %formalname; >
```

```

<!ELEMENT FormalName (#PCDATA) >
<!ATTLIST FormalName %localid;
          Scheme CDATA #IMPLIED >

<!ELEMENT Description (#PCDATA) >
<!ATTLIST Description %localid;
          xml:lang CDATA #IMPLIED
          Variant CDATA #IMPLIED >

```

以下の例では、[TopicSet](#)がEvent（出来事）、Person（人物）、Company（会社）という3つのタイプの[Topic](#)を持つ。これらの[TopicType](#)はすべてIPTC Topic Types vocabularyから引き出される形式名によって識別される。このIPTC Topic Types vocabularyは[Catalog](#)で[TopicType](#)要素のための[default vocabulary](#)であると宣言される。

最初の[Topic](#)はEvent（出来事）で、英語で「Iran-Iraq war（イラン・イラク戦争）」と書かれている。

2番目の[Topic](#)は「Tony Blair（トニー・ブレア）」と記述されたPerson（人物）についてである（その[Description](#)要素値に関連づけられた特定の言語はない）。さらにこのPerson（人物）についての[Details](#)属性が、外部ファイル「whoswho.xml」の中でtonyblairとブックマークされた場所で見つけることができる。

あとの2つの[Topic](#)はCompany（会社）についてであり、より形式的に識別されている。それらはそれぞれCompany Name（会社名）という特定の[Variant](#)属性を持つ[Description](#)を持つ。さらに、それぞれが2つの[FormalName](#)を持ち、そのうち1つはRIC naming schemeに、もう1つがNASDAQ naming schemeに属している。

```

<?xml version="1.0"?>
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20001006:NewsMLv1.0:1"
"http://www.iptc.org/NewsML/DTD/NewsMLv1.0.dtd">
<NewsML>
  <Catalog>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:IptcTopicTypes:1</Urn>
      <Url>http://www.iptc.org/NewsML/topicsets/iptc-topicatypes.xml</Url>
      <DefaultVocabularyFor Context="TopicType"/>
    </Resource>
  </Catalog>
  <TopicSet FormalName="Event"/>
    <Topic Duid="event1">
      <TopicType FormalName="Event"/>
      <Description xml:lang="en-GB">Iran-Iraq war</Description>
    </Topic>
    <Topic Duid="person1" Details="whoswho.xml#tonyblair">
      <TopicType FormalName="Person"/>
      <Description>Tony Blair</Description>
    </Topic>
    <Topic Duid="company1">
      <TopicType FormalName="Company"/>
      <FormalName Scheme="RIC">DELL.O</FormalName>
      <FormalName Scheme="NASDAQ">DELL</FormalName>
      <Description Variant="Company Name">Dell Computer</Description>
    </Topic>
    <Topic Duid="company2">
      <TopicType FormalName="Company"/>
      <FormalName Scheme="RIC">RTRSY.O</FormalName>
      <FormalName Scheme="NASDAQ">RTRSY</FormalName>
      <Description Variant="Company Name">Reuters</Description>
    </Topic>
  </TopicSet>
  ...
</NewsML>

```

以下の例では、IPTCサブジェクト・コード vocabularyが、[TopicSet](#)内の[TopicSetRef](#)要素を通じた参照によって含

まれる。また、1つの追加の[Topic](#)要素も与えられる。これはIPTC topic types naming schemeによって定義される「[SubjectMatter](#)」の[TopicType](#)を持つ。その追加の[Topic](#)には「Building Design」という短い英語の記述と、「The art and science of designing buildings」というフルの英語記述がある。それにはまた2つの[FormalName](#)が与えられている。IptcSubjectCodes naming schemeではその[FormalName](#)は「01002000」であり、myscheme naming schemeではその[FormalName](#)は「BDES」である。このことは、myscheme naming schemeでの[FormalName](#)「BDES」へのどんな参照も、IPTCサブジェクト・コードvocabularyで「01002000」と名づけられたものとまさに同じものを参照することを意味する。

```
<TopicSet Duid="mysubjects" FormalName="SubjectMatter">
  <TopicSetRef TopicSet="urn:newsml:iptc.org:iptc:20001006:IptcSubjectCodes"/>
  <Topic Duid="mysubject1">
    <TopicType FormalName="SubjectMatter" Vocabulary="urn:iptc:20001006:IptcTopicTypes"
Scheme="IptcTopicTypes"/>
    <FormalName Scheme="myscheme">BDES</FormalName>
    <FormalName Scheme="IptcSubjectCodes">01002000</FormalName>
    <Description xml:lang="en-GB" Variant="ShortDesc">Building Design</Description>
    <Description xml:lang="en-GB" Variant="FullDesc">The art and science of designing
buildings</Description>
  </Topic>
</TopicSet>
```

システムが実際にIPTCサブジェクト・コード vocabularyにアクセスし、システム内にローカルに含まれるTopicsと[IPTC](#)のTopicsをマージするとしたら、結果として以下のように示すようにマージされた[Topic](#)エレメントが生じるだろう。これにより、myscheme でBDESと呼ぶtopicが、[IPTC](#) のvocabularyでは「Architecture (建築)」として記述されていることがわかる。

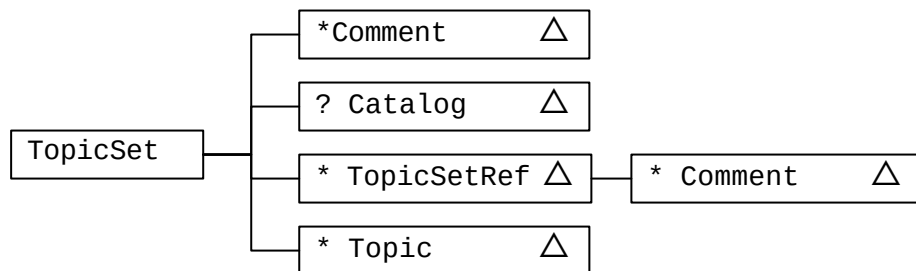
```
<Topic Duid="mergedtopic1">
  <TopicType FormalName="SubjectMatter"/>
  <FormalName Scheme="IptcSubjectCodes">01002000</FormalName>
  <FormalName Scheme="myscheme">BDES</FormalName>
  <Description xml:lang="en-GB" Variant="ShortDesc">Building
Design</Description>
  <Description xml:lang="en-GB" Variant="FullDesc">The art and science of
designing buildings</Description>
  <Description xml:lang="en-GB" Variant="Name">Architecture</Description>
</Topic>
```

上記の手法は、ほかの[controlled vocabulary](#)から引き出された用語と1つの[controlled vocabulary](#)の中で用語が同義であることを表明するための、一般的なメカニズムとして使用することができる。このメカニズムの使用を容易にするために、コントロールされたcontrolled vocabulariesとしての使用を意図される[TopicSet](#)においてすべての[FormalName](#) (要素) に[Scheme](#)属性を含むのは良い方法である。

TopicSet要素

[TopicSet](#)は[Topic](#)の集まりである。NewsML には[Topic](#)要素と[Topic](#)属性があるので、注意が必要である。[TopicSet](#)要素の意味は[FormalName](#)属性で表される。この[FormalName](#)属性は単に[TopicSet](#)のタイプを（形式的に）表すものであり、NewsML 文書中における重要度は低い。[TopicSet](#)には[Topic](#)下位要素の他に、補助的に下位要素として[Comment](#)、[Catalog](#)、[TopicSetRef](#)を持つ。[TopicSet](#)は宣言であって、他の要素で[TopicSet](#)内の該当[Topic](#)をポイントして使用する。

[ツリー図]



[TopicSet](#)は2つの使い方がある。1つは、実世界事情（トピック）の集合を表す。これは人、場所、会社、あるいはある重要な意味をもつその他のものであるかもしれない。もう1つは、[Controlled Vocabulary](#)である。

※仕様書では実世界事情（トピック）に[controlled vocabulary](#)が含まれているが、解説書ではこれを区別して説明することとする。

[controlled vocabulary](#)（コントロールド・ボキャブラリ）

[controlled vocabulary](#)（コントロールド・ボキャブラリ）とは、（一般的には）[Scheme](#)属性と[FormalName](#)属性の組み合わせで、[TopicSet](#)内の唯一の[Topic](#)を示すようにコントロールされたボキャブラリのことである。つまり、[FormalName](#)下位要素を持つ[Topic](#)のことである。

[controlled vocabulary](#)では、[Topic](#)が使用する候補値の1つに対応し、[TopicSet](#)が値の候補の集合（ボキャブラリ）に対応する。通常、[controlled vocabulary](#)では[TopicType](#)下位要素は1つである。

実世界事情（トピック）を外部[TopicSet](#)ファイルで管理する場合など、[controlled vocabulary](#)として扱えるように[FormalName](#)下位要素を持たせることもある。しかし、実世界事情（トピック）をポイントする要素では[FormalName](#)属性を使用できない。[controlled vocabulary](#)は、[FormalName](#)属性で[Topic](#)をポイントするような要素で利用される。

[TopicSet](#)要素が記述できる個所

[TopicSet](#)は、NewsML 文書中のいくつかの個所に記述できる。NewsML 要素の中と[NewsComponent](#)要素の中で記述された場合は、各サブツリーの中でのみ有効な[TopicSet](#)として使用される。[NewsItem](#)要素の中で記述された場合、外部参照される[TopicSet](#)ファイルを作成するために使用される。これによって作成された NewsML ファイルは通常、[controlled vocabulary](#)として使用される。なぜならば、[controlled vocabulary](#)ではない[Topic](#)をポイントするための[Topic](#)属性は[TopicUse](#)、[TopicOccurrence](#)、[Party](#)各要素に存在するが、参照範囲が同じ文書内に限られるからである。

TopicSet 要素が記述できる個所 (1)

次の例は、通常の ([NewsComponent](#)がある) NewsML ファイルに[TopicSet](#)要素を記述できる個所を表している。

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

TopicSet 要素が記述できる個所 (2)

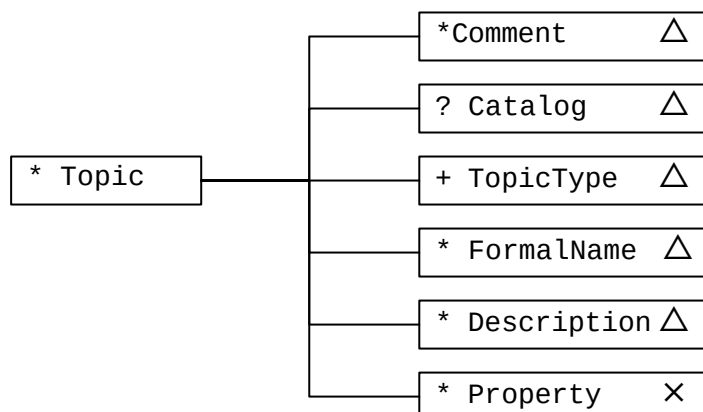
次は、外部参照される[TopicSet](#)ファイルで、[TopicSet](#)要素を記述する個所を表している。

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

Topic要素

[Topic](#)要素は、実世界事情（トピック）あるいは、[controlled vocabulary](#)の1つを表す。[Topic](#)要素は[Details](#)属性を持つことができ、これによって http URLや NewsML URNの形でその[Topic](#)に関する追加情報をポイントする。[TopicType](#)下位要素は、[Topic](#)の種別（タイプ）を表す。[Description](#)下位要素には[Topic](#)の説明が記述される。[FormalName](#)下位要素があれば[controlled vocabulary](#)としても使用できる。[Topic](#)要素は他に、補助的な下位要素として[Comment](#), [Catalog](#), [Property](#)を持つ。

[ツリー図]



[Topic](#)要素は[FormalName](#)下位要素を持つタイプと持たないタイプの2つに分類される。NewsML には、[FormalName](#)要素と[FormalName](#)属性があるので、注意が必要である。

[Topic](#)要素で[FormalName](#)下位要素を持たないものは、実世界事情（トピック）を表現する。また、1つ以上の[TopicType](#)下位要素を持つ場合が多くあると推測できる。

実世界事情（トピック）を指定する方法は、[TopicUse](#), [TopicOccurrence](#), [Party](#)各要素に含まれる[Topic](#)属性に#記号と、該当[Topic](#)の[Duid](#)を続けて構成するポインタで指定する。

実世界事情（トピック）を指定する例

```
<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>
```

[controlled vocabulary](#)の1つの値を指定する方法は、対象要素に含まれる[FormalName](#)属性に該当[Topic](#)の[FormalName](#)下位要素の内容を記述する。より明確に[Topic](#)を指定するために、[naming scheme](#)を指定することもできる。[naming scheme](#)の指定方法は対象要素の[Scheme](#)属性に、該当[Topic](#)の[FormalName](#)下位要素に含まれる[Scheme](#)属性の値を指定する。

controlled vocabulary を指定する例

```
<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-JP">コンテンツ単価</Description>
</Topic>
<Topic Duid="topic.nskprice">
  <TopicType FormalName="Property" Scheme="IptcTopicType"/>
  <FormalName Scheme="NskProperty">NSK:PriceYEN</FormalName>
  <Description xml:lang="ja-JP">価格（円）</Description>
</Topic>
</TopicSet>
<Metadata>
  <Catalog>
    <Resource>
      <Url>#LocalTopicSet</Url>
      <DefaultVocabularyFor Context="MetadataType"/>
      <DefaultVocabularyFor Context="Property"/>
    </Resource>
  </Catalog>
  <!--コンテンツ単価の情報-->
  <MetadataType FormalName="NSK:ContentPrice"/>
  <!--このコンテンツ単価（円）は 1000 -->
  <Property FormalName="NSK:PriceYEN" Scheme="NskProerty" Value="1000"/>
</Meatadata>
...
</NewsComponent>

```

TopicSetRef要素

あるTopicSetから他の既存のTopicSetを引用したい場合、TopicSetRef要素を使用する。含めるTopicをもつ他のTopicSetへのポインタを、TopicSetRef要素のTopicSet属性で指定する。このポインタは、内部あるいは外部のTopicSetを識別する http URL または NewsML URN か、現在の文書中にあるTopicSetのDuid属性の値があとに続く # 記号からなるfragment identifierのいずれかである。

TopicSetRef要素を使用することによって、参照元のTopicSetに、ポイントされているTopicSetのTopicをすべて含めることができる。

TopicSetRef を使って Topic を追加する例

つぎの例は、IPTCのProviderに NSK（日本新聞協会）を追加している。

```
<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-JP">日本新聞協会</Description>
    </Topic>
  </TopicSet>
  <AdministrativeMetadata>
    <Provider>
      <Party FormalName="NSK" Scheme=" NskTiffServiceId"/>
    </Provider>
    <Creator>
      <Party FormalName="IPTC" Scheme="IptcProvider"/>
    </Creator>
  </AdministrativeMetadata>
  ...
</NewsComponent>
```

すでに定義されているTopicを拡張する場合も、TopicSetRef要素が使用できる。TopicSetRef要素によって含まれるTopicと、新しいTopicSetのTopicにおいて、同じFormalNameとSchemeの組み合わせがある場合、これら2つのTopicが同じものを指していることになる。このようなケースでは、2つのTopicを合成（マージ）すると解釈すればよい。Topicのマージは、システムによって物理的に実行される必要はないが、マージされたように解釈する必要がある。

TopicSetRef を使って Topic をマージする例

つぎの例は、IPTCのProviderに NSK（日本新聞協会）と KYODO NEWS（共同通信社）を追加している。共同通信社についてはIPTCが既に定義した KYODO と同じ Topic にしている。このため、KYODO の Topic も KYODO NEWS として扱われることを意味している。（別の表現をすると、KYODO NEWS も KYODO と同じ処理をすることが期待される。）

```
<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="IptcProvider.KYODO">
    <TopicType FormalName="Provider" Scheme="IptcTopicType"/>
    <FormalName Scheme="NskTiffServiceId">KYODO NEWS</FormalName>
    <FormalName Scheme="IptcProvider">KYODO</FormalName>
    <Description xml:lang="ja-JP">共同通信社</Description>
  </Topic>
  <Topic Duid="NskProvider.NSK">
    <TopicType FormalName="Provider" Scheme="IptcTopicType"/>
    <FormalName Scheme="NskTiffServiceId">NSK</FormalName>
    <Description xml:lang="ja-JP">日本新聞協会</Description>
  </Topic>
</TopicSet>
...
</NewsComponent>

```

サンプル TopicSet の説明

上記のサンプル TopicSet をポイントするには、つぎのように記述する。

(1) [FormalName](#)属性でポイントする場合

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

(2) [Topic](#)属性でポイントする場合

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

(通常は、[Provider](#)の[Topic](#)を[TopicOccurrence](#)でポイントすることはない)

A Topic/@Duid	B Topic/FormalName	C Topic/FormalName/@Scheme	D Topic/TopicType	補足
NskProvider.NSK	NSK	NskTiffServiceId	Provider	
IptcProvider.KYODO	KYODO NEWS	NskTiffServiceId	Provider	同一 Topic
	KYODO	IptcProvider	Provider	同一 Topic
:	:	:	:	

TopicType要素

[Topic](#)は必ず[TopicType](#)下位要素を持つ。[TopicType](#)要素で[Topic](#)の種別を表す。1つの[Topic](#)が複数の[TopicType](#)下位要素を持つ場合もある。[Topic](#)の種別は、[TopicType](#)要素の[FormalName](#)属性に、[Topic](#)の種別のために用意された[controlled vocabulary](#)の値を設定することによって表す。

TopicType の例

```
<Topic Duid="topic.nskstdnewsinfomation">
  <TopicType FormalName="MetadataType" Scheme="IptcTopicType"/>
  <FormalName Scheme="NskMetadataType">NskMetadata</FormalName>
  <Description xml:lang="ja-JP">N S K拡張用メタデータ</Description>
</Topic>
```

[Topic](#)の種別のための[controlled vocabulary](#)には、IPTCの[TopicType](#)を記述した[TopicSet](#)ファイル(urn:newsml:iptc.org:20001006:topicset.iptc-topictype)もしくは、このファイルを[TopicSetRef](#)でマージした[TopicSet](#)を使う。これらの[controlled vocabulary](#)の中にある[Topic](#)の種別は"[TopicType](#)"である。

```
urn:newsml:iptc.org:20001006:topicset.iptc-topictype:1 の抜粋
<TopicSet Duid="newsmltopicitypes" FormalName="TopicType">
  ...
  <!--Topicの種別"MediaType"を定義している-->
  <Topic Duid="TopicTypes.NewsML.MediaType">
    <TopicType FormalName="TopicType"/>
    <FormalName Scheme="IptcTopicType">MediaType</FormalName>
  </Topic>
```

[Topic](#)の種別に、IPTCが定義した値を使用することによって、ある程度の自動的な処理が期待できる。例えば、"Person"の[Topic](#)の種別に属する[Topic](#)は人物データベースに、"Event"の[Topic](#)の種別に属する[Topic](#)はイベントデータベースに自動登録するシステムを作ることができる。

[controlled vocabulary](#)の場合は[Topic](#)の種別だけでなく、[FormalName](#)そのものもIPTCに定義した値を使用することによって、共通の制御を表すことができるものがある。[Topic](#)の種別が"[Status](#)", "[NewsItemType](#)"である[Topic](#)は、その例である。IPTCの[FormalName](#)に独自の意味を与えてはならない。もし、独自の制御を指定したい場合は、[Topic](#)を追加して、その[FormalName](#)を利用する。追加した[Topic](#)に対する制御は配信側と受信者側の間で事前に理解されている必要がある。

FormalName要素

[FormalName](#)要素は、[Topic](#)をポイントする際の名称を定義する。[FormalName](#)要素は、[Scheme](#)属性を持つことができる。NewsML には、[FormalName](#)要素と[FormalName](#)属性があるので注意が必要である。

[FormalName](#)要素の[Scheme](#)属性は、任意の名前型(naming scheme)を定義して、これを[FormalName](#)に割り当てることができる。[FormalName](#)要素の[Scheme](#)属性は、他の要素の[Scheme](#)属性と仕様が異なるので注意が必要である。

(他の要素の[Scheme](#)属性は、この[FormalName](#)要素の[Scheme](#)属性によって割り当てられた既存の名前型を指定するものである。これに対して、[FormalName](#)要素の[Scheme](#)属性は名前型を作り出す。)

[FormalName](#)要素の内容と、[Scheme](#)属性値の組み合わせは、[TopicSet](#)の中で唯一(ユニーク)でなければならない。[Topic](#)要素には、複数の[FormalName](#)下位要素を持つことができる。これにより複数の[FormalName](#)と[Scheme](#)属性値の組み合わせが1つの[Topic](#)をポイントすることもある。

FormalName要素を複数持つ例

```
<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-JP">共同通信社</Description>
</Topic>
```

Description要素

オプションの [Description](#) 要素は、[Topic](#) の説明を表す。複数の記述が可能である。xml:lang 属性は、その説明がどの言語で記述されているかを示す。Valiant 属性は同じ言語で複数の説明を記述するときに、それらを区別するために使用される。

Description 要素を複数持つ例 (topicset.nsk-party.xml より)

```
<Topic Duid="Nsk.Party.Asahi">
  <TopicType Scheme="NskTopicType" FormalName="Party"/>
  <FormalName Scheme="NskTiffServiceId">ASAHI</FormalName>
  <FormalName Scheme="NskParty">The_Asahi_Shimbun_Company</FormalName>
  <Description xml:lang="ja-JP" Valiant="Name">株式会社 朝日新聞社</Description>
  <Description xml:lang="en-GB" Valiant="Name">The Asahi Shimbun Company
  </Description>
  <Description xml:lang="ja-JP" Valiant="ShortName">朝日新聞社</Description>
  <Description xml:lang="en-GB" Valiant="ShortName">Asahi</Description>
</Topic>
```

[NskNewsML:1 記述形式]

	使用許可	備考
NewsML/TopicSet	△	
NewsItem/ TopicSet	○	TopicSet ファイルを作成するために使用する
NewsItem/NewsComponent/TopicSet	△	
TopicSet/@FormalName	○	
TopicSet/@Scheme	△	
TopicSet/@Vocabulary	△	
TopicSet/Topic	○	
Topic/TopicType	○	
Topic/FormalName	○	
Topic/FormalName/@Scheme	○	
Topic/Description	○	
Topic/Property	×	
TopicType/@FormalName	○	
TopicType/@Scheme	△	

コラム：“Topic は NewsML のキモ”

[XML](#)は要素の値を[DTD](#)では規定できません。属性値に関しては[DTD](#)で規定することも可能ですが、値を変更、追加するたびに[DTD](#)を変えなければならなくなります。

これらに対応するために[XML](#)スキーマの実装が待たれるところです。

要素という入れ物は用意したが、その用途となかに入っている値が各社ばらばらでは完全なデータ交換はできません。NewsMLでは、これを“かなり”回避するために値を規定する仕組みがあります。これを参照することにより、各社共通の値を使うことができます。

値を規定する為に使用するのが、
“[Topic](#)”（トピック）です。

一つの事柄について、一つの“[Topic](#)”で表現します。

“[Topic](#)”（トピック）には、一つの事柄を表す値を示す[FormalName](#)要素、内容の説明が記述されている[Description](#)要素等が用意されており、NewsML 文書内で使用する要素や属性値の値とそれのもつ意味合いが規定できます。

NewsML が共通フォーマットとして機能するために用意された、“NewsML のキモ”です。
この[Topic](#)を列挙してまとめたのが、
“[TopicSet](#)”（トピックセット）です。

各要素や、属性ごとに使用できる値の範囲をまとめるなど、“セット”とすることで意味合いを持たせることができます。値の一覧として利用する場合には、これを正式には“[Controlled Vocabulary](#)”と呼びますがより一般的に

“ボキャブラリ”と呼びます。

このボキャブラリを NewsML 文書内から、[URL](#)や[URN](#)によって参照する要素名が、
“[Catalog](#)”（カタログ）です。

[Catalog](#)では、各要素ごとのボキャブラリを[Resource](#)要素を使って参照先を記述します。また、この[Catalog](#)自体を[Href](#)属性を使って参照することもできます。

この際に、必要な[Resource](#)要素をすべて集めて別ファイルとしたのが、
“マスターカタログ”です。

常に NewsML 文書内に長い[Catalog](#)を書かないで済むのでデータが小さくなる、更新したいときも一つの“マスターカタログ”を変更すれば良いなどの利点があります。

また、利用社間において、同一のマスターカタログを使用すれば、より共通フォーマットとしての意味合いが強くなります。

[IPTC](#)ではマスターカタログは NewsML の形を取っていますので、NskNewsML:1 も同じ形で使用することになりましたが、使用社の使いたい要素、ボキャブラリによってマスターカタログは異なるため、各社共通な万能マスターカタログは存在しません。

NskNewsML:1 では、マスターカタログを使用する場合は、利用社間で作成することを前提として、マスターカタログを作成するために利用可能な[Resource](#)要素を列記するだけにしました。これをマスターカタログと区別するために、

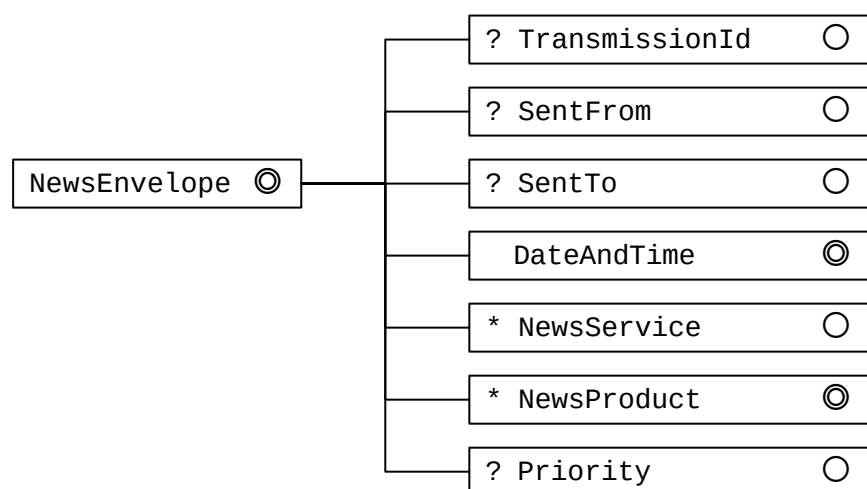
“リソースセット”と呼びます。

5.4 NewsEnvelope

[NewsEnvelope](#) 要素は NewsML 文書がビジネス・ワークフローやニュース配信社と受信社間の契約関係の中でどのように使われるかについての情報を記述するためのものであり、1つの NewsML 文書に必ず1つだけ存在する。この要素は最低限1つの[DateAndTime](#) 要素を含まねばならない。それに加えて、1つの[TransmissionId](#)、[SentFrom](#)、[SentTo](#)、[Priority](#)要素と1つ以上の[NewsProduct](#)、[NewsService](#)要素の両方またはいずれかを含んでもよい。

```
<!ELEMENT NewsEnvelope (TransmissionId?, SentFrom?, SentTo?, DateAndTime, NewsService*, NewsProduct*, Priority?)>
<!ATTLIST NewsEnvelope %localid; >
```

[ツリー図]



[NSK 解説]

[NewsEnvelope](#)とは送信時刻や送信 ID、送信元、受信先など NewsML 文書全体の送受信に関わる情報を記述する場所であり、封筒のあて名の役目をする。

[NskNewsML:1 記述形式]

NewsEnvelope 要素 ..◎

[例]

```
<NewsEnvelope>
  <TransmissionId>20010313provider0001:2+20010222:0001</TransmissionId>
  <SentFrom>
    <Party FormalName="A-COMPANY"/>
  </SentFrom>
  <SentTo>
    <Party FormalName="B-COMPANY"/>
  </SentTo>
  <DateAndTime>20010313T1500+0900</DateAndTime>
```

```
<NewsProduct FormalName="NskNewsML:1"/>  
<Priority FormalName="5"/>  
</NewsEnvelope>
```

5.4.1 TransmissionId

[TransmissionId](#)は NewsML 文書伝送のための識別子である。この要素は同じ配信社からの異なった伝送すべてにおいて唯一でなければならない。仮に送信側が、受信が正常に行われたことに確信がもてなく伝送を繰り返したい時は、同じ内容の[TransmissionId](#)が使われてもよい。しかし、最初の伝送と二度目とを区別するために [Repeat](#)属性を提供しなければならない。 [Repeat](#)属性の値がとる形式は配信社が決める。同様に[TransmissionId](#)自体のフォーマットも配信社の決定事項である。たとえばチャンネル識別子の後にシーケンス番号が続くといった形式が考えられる。

```
<!ELEMENT TransmissionId (#PCDATA) >
<!ATTLIST TransmissionId %localid;
                Repeat CDATA #IMPLIED >
```

```
<TransmissionId Repeat="second attempt">abc123</TransmissionId>
```

[NSK 解説]

通信レベルの再送時は変更しなくてもよい。ユーザー要求による再送時は[TransmissionId](#)を変更するか、変更しない代わりに[Repeat](#)属性をつける。同じ相手で[TransmissionId](#)が同じ場合は同じ NewsML 文書が送られたことを示す。

[TransmissionId](#)を唯一にするために [NewsItemId](#)や [PublicIdentifier](#) を使用してはならない。[NewsItemId](#)や[PublicIdentifier](#)は[NewsItem](#)の識別子であり、伝送の識別子にはならないからである。例えば[TransmissionId](#)に[NewsItemId](#)を使用した場合、[NewsItem](#)の改版が起きると[NewsItemId](#)は変わらないが、[TransmissionId](#)は変更して送らないと[TransmissionId](#)の重複が発生する。よって使用してはならないのである。

[NskNewsML:1 記述形式]

NewsEnvelope/TransmissionId 要素 ・○

5.4.2 SentFrom と SentTo

[SentFrom](#)要素は、NewsML 文書を送信した 1 つ以上の組織を識別し、[SentTo](#)要素はその文書を送られた 1 つ以上の組織を識別する。これら 2 つの内容モデルは party エンティティによって提供され、ニュース・ワークフローにおいて特定の役割を果たす個人、組織、企業を記述する。オプションの [Comment](#) 要素は、自然言語で非形式的な追加情報を提供する。[Comment](#)要素はオプションとして [xml:lang](#)属性と TranslationOf 属性を持つ。[xml:lang](#)属性は [XML](#) 要素の内容の言語を識別する。それは [XML](#) の仕様に定義されており、その値は [IETF](#) RFC3066 で定義されたものでなければならない。[IETF](#) RFC3066 では、ISO639-1 か ISO639-2 の言語コードと、ISO3166-alpha2 か ISO3166-alpha3 の国コードの任意の仕様（言語コードと国コードをハイフンで組み合わせること）が許されているが、（NewsML では言語コードは）ISO639-1 の 2 文字言語コードを使う必要がある。発行者は 2 文字または 3 文字の国コードを選ぶ。そのために、[xml:lang](#)属性は ll-CC か ll-CCC の構造になる。TranslationOf 属性は、この [Comment](#)が直訳となる他の [Comment](#)要素へのポインタである。コメントタイプは、[Comment](#)要素の [FormalName](#)属性で任意に命名される。[Comment](#)要素の [Vocabulary](#)属性は、[FormalName](#)の意味を定義する [controlled vocabulary](#)へのポインタである。[Scheme](#)属性は、もし存在すれば、ボキャブラリ内のどの [naming scheme](#)がこの形式名に当てはまるかを識別する。

その [Property](#)要素、またはその [FormalName](#)属性、[Vocabulary](#)属性、[Scheme](#)属性を通して、[Party](#) 要素は当該の組織である [Topic](#)を識別する。オプションの [Topic](#)属性はその [Topic](#)への直接のポインタとして使用できる。そのポインタは http の URL 形式、または NewsML の URN 形式、または #記号の後に現行

文書での[Topic](#)要素の[Duid](#)属性の値を続ける形をとる。

```
<!ENTITY % party " (Comment* , Party+ )">

<!ELEMENT SentFrom (%party;)>
<!ATTLIST SentFrom %localid; >

<!ELEMENT SentTo (%party;)>
<!ATTLIST SentTo %localid; >

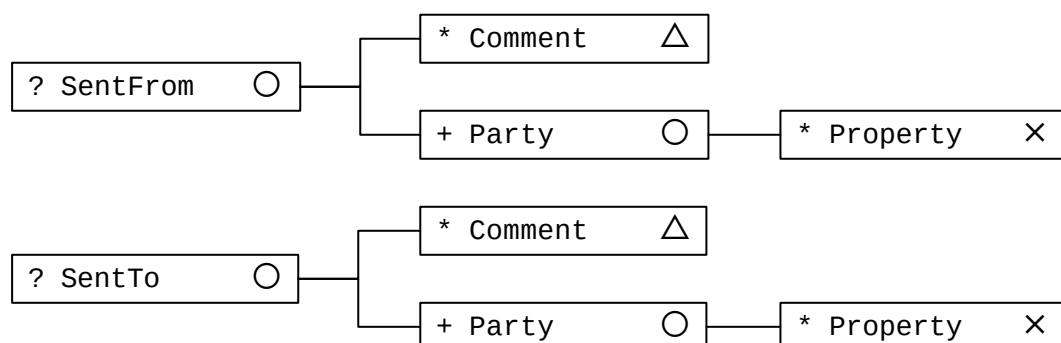
<!ELEMENT Comment (#PCDATA)>
<!ATTLIST Comment %localid;
               xml:lang CDATA #IMPLIED
               TranslationOf IDREF #IMPLIED
               FormalName CDATA #IMPLIED
               Vocabulary CDATA #IMPLIED
               Scheme CDATA #IMPLIED >

<!ELEMENT Party (Property)*>
<!ATTLIST Party %localid;
               %formalname;
               Topic CDATA #IMPLIED >
```

以下の例は、文書を送信している[Party](#)の、MyCompanyCodes [controlled vocabulary](#)におけるxyz naming schemeでの形式名が、MYCODEであることを表している。[Party](#)要素の[Vocabulary](#)属性は、MYCODEの意味を解析するために使用される[controlled vocabulary](#)を提供する[TopicSet](#)を識別する。

```
<SentFrom>
<Party FormalName="MYCODE" Scheme="xyz"
  Vocabulary="urn:newsml:mycompany.com:20010101:MyCompanyCodes:1"/>
</SentFrom>
```

[ツリー図]



[NSK 解説]

[SentFrom](#)要素には送信元、[SentTo](#)要素には送信先の組織を表す。

[Comment](#)要素には自然言語で自由な形式で追記情報を記述が可能である。ただし、NewsML1.1 より [xml:lang](#)属性が ISO639-1 と ISO3166-alpha2 の組み合わせ(II-CC)、または ISO639-1 と ISO3166-alpha3 の組み合わせ(II-CCC)で表現することになったため注意が必要である。日本語を表現する場合は、II-CC だと「xml:lang="ja-JP"」となり、II-CCC は「xml:lang="ja-JPN"」になる。また、国コードはかつて[IPTC](#)

により[TopicSet](#)として管理されていた。しかしこの国コードの[TopicSet](#)は、事実上、ISO3166 と ISO639 で定義されているものをそのまま使用しているだけなので意味をなさない。それゆえ、現在、これらの ISO 仕様の値あるいは言語のための RFC3066 に一致した値を採用することや、ISO あるいは RFC の web サイトの情報ページに示されているボキャブラリの値を加えることを認めている。こういった処理は、もはや[IPTC](#)が iso-country-codes と iso-language-code の[TopicSet](#)を管理しないため、[IPTC](#)も強く推奨している。

[Party](#)要素は組織の名称を表す。内容は、送信元と送信先の組織間で取り決めを行う必要がある。また、NewsML1.1 より[Party](#)の子要素として[Property](#)要素が追加されている。

下位要素の[Comment](#)要素の値は人間が読むことを想定した形式で記述される。[Party](#)要素の直前に[Comment](#)要素を記述して、子要素の[Party](#)要素を解説する。

なお、[Comment](#)要素はオプションとして[xml:lang](#)属性と TranslationOf 属性を持つ。[xml:lang](#)属性で[Comment](#)要素値の言語を指定することにより、さまざまな言語で記述することができる。

TranslationOf属性は IDREF 型で、同一文書内の同じ ID 型属性値をとる[Comment](#)要素を示すポイントとなる。

[NewsML 仕様の変更点：NewsML v1.1]

1. [Comment](#)要素の[xml:lang](#)の表現方法が従来は言語コードのみであったが、言語コードと国コードの組み合わせに変更された。
2. [Comment](#)要素に[FormalName](#)、[Vocabulary](#)、[Scheme](#)が追加された。
3. [Party](#)要素に[Property](#)子要素が追加された。

[NskNewsML:1 記述形式]

Party のボキャブラリは新聞社・通信社間の場合、NSK オリジナルボキャブラリを推奨するが、配信社と受信社で決める。

NewsEnvelope/SentFrom 要素 …… ○

NewsEnvelope/SentFrom/Party 要素 ○

NewsEnvelope/SentTo 要素 …… ○

NewsEnvelope/SentTo/Party 要素 …… ○

```
<SentFrom>
  <Party FormalName="A-company"/>
</SentFrom>
<SentTo>
  <Party FormalName="B-company"/>
</SentTo>
```

[NskNewsML:1.2 記述形式]

NewsEnvelope/SentFrom/Comment △

NewsEnvelope/SentFrom/Comment/@FormalName △

NewsEnvelope/SentFrom/Comment/@Vocaburaly △

NewsEnvelope/SentFrom/Comment/@Scheme △

NewsEnvelope/SentTo/Comment	△
NewsEnvelope/SentTo/Comment/@FormalName	△
NewsEnvelope/SentTo/Comment/@Vocaburaly	△
NewsEnvelope/SentTo/Comment/@Scheme	△
NewsEnvelope/SentFrom/Party/Property 要素	×
NewsEnvelope/SentTo/Party/Property 要素	×

NewsML v1.1 で追加された Comment 要素の FormalName 属性、Vocaburaly 属性、Scheme 属性は使用しても良いが、同じく追加された Party 要素の Propertr 子要素は使用してはいけない。

Comment 要素の [xml:lang](#) の表現方法は、言語コードと国コードの組み合わせで表現する。

5.4.3 DateAndTime

[DateAndTime](#) 要素は、伝送時の日付とオプションとしての時間を含む。これは、ISO 8601:2000(E)フォーマットで規定されているとおり、日付は CCYYMMDD 形式で表現する。続けてオプションとして文字 T と現地時間を HHMMSS 形式、さらにオプションとして+、-に続けて現地時間と協定世界時([UTC](#))との時差を HHMM 形式で表現する。時差が+0000 のところでは添え字“Z”が代わりに使える。

```
<!ELEMENT DateAndTime (#PCDATA)>
<!ATTLIST DateAndTime %localid; >
```

以下の例は、この[NewsItem](#)は現地時間における 2000 年 10 月 6 日 14 時に送られたものであり、協定世界時([UTC](#))より 2 時間早いことを示している。

```
<DateAndTime>20001006T1400+0200</DateAndTime>
```

[UTC](#)表記で、時差がない（例えば 2002 年 3 月 6 日 12 時）場合、代わりに添え字を使うことができる。

```
<DateAndTime>20020306T120000Z</DateAndTime>
```

[NSK解説]

ISO8601の書式がすべて許されているのではなくCCYYMMDD形式のみである。オプションの時間はHHMMSS形式のみで、協定世界時との時差はHHMM形式のみある。ユーザーが明示的に再送を要求したときには別の値となる。通信レベルでの再送については、同一の値にする。

[参考： ISO8601記述形式]

- ・ 暦日付（こよみひづけ）の完全表記： CCYYMMDD 例) 20010313
- ・ 時刻のみの省略表記： HHMMSS 例) 150000
- ・ 現地時間と協定世界時との差（時差）を付けた場合： HHMMSS±HHMM 例) 150000+0900
- ・ 日付と時刻の組み合わせの表記： 時刻の先頭に文字“T”を指示記号として付ける。

※ISO8601は、1988年に発行された国際規格で、この様式に沿って一部記述を追加したものが JIS X 0301 として1992年に発行されている。

[NskNewsML:1記述形式]

NSKではつぎのいずれかの形式にする。日本時刻の協定世界時との差は+0900である。

CCYYMMDD

CCYYMMDDThhmmss±hhmm

NewsEnvelope/DateAndTime 要素 ……◎

5.4.4 NewsService と NewsProduct

[NewsService](#)要素と[NewsProduct](#) 要素はそのパッケージが含まれているサービスやプロダクトを示している。複数の[NewsService](#)要素と[NewsProduct](#)要素が許可されている。[FormalName](#)属性の値は、サービスまたはプロダクトの形式名である。その意味と許される値は、[Vocabulary](#)属性と[Scheme](#)属性で識別される[controlled vocabulary](#)により決定される。

```
<!ELEMENT NewsService EMPTY>
<!ATTLIST NewsService %localid;
                %formalname; >
<!ELEMENT NewsProduct EMPTY>
<!ATTLIST NewsProduct %localid;
                %formalname; >
```

以下の例は、スポーツ (SPORTS) サービス、一般情報 (GENERALINTEREST) サービスと、WebWireプロダクトに属するパッケージを表している。スポーツと一般情報という用語は、MyPressCompany's Servicesのvocabularyから引け、WebWireという用語はMyPressCompany's Productsのvocabularyから引ける。

```
<NewsML>
  <Catalog>
    <Resource Vocabulary="urn:newsml:iptc.org:20001006:IptcPriority:1"
      <Urn>urn:newsml:mpc.com:20010101:MpcServices:1</Urn>
      <DefaultVocabularyFor Context="NewsService"/>
    </Resource>
    <Resource>
      <Urn>urn:newsml:mpc.com:20010101:MpcProducts:1</Urn>
      <DefaultVocabularyFor Context="NewsProduct"/>
    </Resource>
  </Catalog>
  <NewsEnvelope>
    <DateAndTime>20001225T1200+0100</DateAndTime>
    <NewsService FormalName="SPORTS"/>
    <NewsService FormalName="GENERAL INTEREST"/>
    <NewsProduct FormalName="WebWire"/>
  </NewsEnvelope>
  ...
</NewsML>
```

[NskNewsML:1 記述形式]

NewsEnvelope/NewsService 要素 …… ○

NewsEnvelope/NewsProduct 要素 …… ◎で繰り返しなし

NskNewsML レベル 1 に沿った NewsML 文書であることを識別するために **NewsProduct** 要素を利用

する。**NewsProduct** 要素は本来、省略可、繰り返しありだが、NSK ガイドラインでは◎で繰り返しなし。ボキャブラリは NSK オリジナルボキャブラリを使用する。

(例) <NewsProduct FormalName="NskNewsML:1"/>

NewsService 要素の **FormalName** 属性値とボキャブラリは各社自由。

[NskNewsML:1.2 記述形式]

NewsProduct 要素は、以下と記述する（尚、使用制限は前バージョンと同じ）

(例) <NewsProduct FormalName="NskNewsML:1.2"/>

5.4.5 Priority

Priority 要素は **NewsItem** の優先順位を示す。**FormalName** 属性の値はその優先順位の形式名である。その意味と許される値は、**Vocabulary** 属性と **Scheme** 属性で識別される **controlled vocabulary** によって決定される。

```
<!ELEMENT Priority EMPTY>
<!ATTLIST Priority %localid;
                %formalname; >
```

この例では、**Priority** の値が IptcPriority vocabulary における「5」とであると宣言されている。

```
<Priority FormalName="5" Vocabulary="urn:newsml:iptc.org:20001006:IptcPriority:1"
Scheme="IptcPriority"></Priority>
```

[NSK 解説]

Priority 要素は送信の優先度を設定する場合に指定する。

[NskNewsML:1 記述形式]

NewsEnvelope/Priority 要素 …………… ○

FormalName のボキャブラリは **IPTC** のボキャブラリを使用する。**FormalName** 属性値は 1（高）～ 5（normal）～ 8（低）で、9 はユーザー用である。

(例) <Priority FormalName="3" />

5.4.6 メタデータの割り当て

[assignment](#)エンティティは[AssignedBy](#)属性、[Importance](#)属性、[Confidence](#)属性、[HowPresent](#)属性、[DateAndTime](#)属性からなる。

[AssignedBy](#)属性はメタデータを割り当てている団体を識別する。それは非形式的に団体を示す文字列（たとえば人の名前）か、#記号に続いてその団体に対応する[Topic](#)の[Duid](#)属性の値からなる[fragment identifier](#)書式のポインタになり得る。

[Confidence](#)属性はメタデータが割り当てられた信用度を示す。[Importance](#)属性はその団体が割り当てたメタデータの重要性を示している。[HowPresent](#)属性はメタデータを適用する方法を示している。これら3つの属性の値は形式名で、その意味はcontrolled vocabulariesによって決められている。それ故、どこで使われるにしてもこれらの属性のそれぞれに適切な default vocabularies を宣言した[Catalog](#)が必要である。さらに、それぞれの[default vocabulary](#)における完全な用語の集合が、対応する属性に許される値の範囲を決定する。もし、[default vocabulary](#)が存在し、[Catalog](#)で示されたリソースが、NewsMLの[TopicSet](#)であるならば、許される値の範囲は正確に[TopicSet](#)における[Topic](#)の集合であることに注意すべきである。

[DateAndTime](#)属性は、メタデータが割り当てられた日付、(オプションとして) 時間を示し、使用されるフォーマットは CCYYMMDDTHHMMSS±HHMM(世紀、年、月、日、時間区切り、時刻、分、秒、タイムゾーン区切り、時間、分)である。これは ISO 8601 で決められた標準フォーマットである。

```
<!ENTITY % assignment " AssignedBy CDATA #IMPLIED
    Importance CDATA #IMPLIED
    Confidence CDATA #IMPLIED
    HowPresent CDATA #IMPLIED
    DateAndTime CDATA #IMPLIED">
```

下記の例は、どのようにメタデータが割り当てられるかを明らかにするため、割り当て属性の使用を示している。[Catalog](#)は、次のことを宣言している。「[Confidence](#)属性のためのデフォルトvocabularyは、IPTC confidence vocabularyにおけるIptcConfidence naming schemeであり、その[URN](#)により識別される。[Importance](#)属性のためのデフォルトvocabularyは、brs.comのウェブサイト上のimportance.xml vocabularyにおけるxyz naming schemeである。[AssignedBy](#)属性のためのデフォルトvocabularyは、[Duid](#)属性値がLocalTopicSetである現在の文書内の[TopicSet](#)におけるcompanycode naming schemeである。」LocalTopicSetの値を持つ[TopicSet](#)はただ1つのTopicを含み、その[TopicType](#)はCompanyでIPTC topic types vocabularyのIptcTopicTypes naming schemeに定義されている。この会社は英語のDescriptionを通じ非形式的にBloomsbury Review Serviceと識別されているが、companycode naming schemeでBRSの[FormalName](#)が与えられている。最終的に、記述メタデータ(descriptive metadata)は協定世界時 (UTC) の2000年12月31日正午にBRS (BRSは上記からBloomsbury Review Serviceとわかる) によって割り当てられ、重要度はbrs.comのウェブサイト上のimportance.xml vocabularyにおいて[FormalName](#)がnormal (普通) と示され、信用度はIPTC confidence vocabularyでHigh (高い) と示されている。これらの設定は、要素・ツリーの下部で明確に再定義されない限り、[DescriptiveMetadata](#)要素の下位要素すべてに適用できる。

```
<NewsML>
  <Catalog>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:IptcConfidence:1</Urn>
      <DefaultVocabularyFor Scheme="IptcConfidence" Context="@Confidence"/>
    </Resource>
    <Resource>
      <Url>http://www.brs.com/vocabularies/importance.xml</Url>
      <DefaultVocabularyFor Scheme="xyz" Context="@Importance"/>
    </Resource>
    <Resource>
      <Url>#LocalTopicSet</Url>
      <DefaultVocabularyFor Scheme="companycode" Context="@AssignedBy"/>
    </Resource>
  </Catalog>
</NewsML>
```

```

</Resource>
</Catalog>
<TopicSet Duid="LocalTopicSet" FormalName="Company">
  <Topic Duid="company1">
    <TopicType FormalName="Company" Scheme="IptcTopicTypes"
Vocabulary="urn:newsml:iptc.org:20001006:IptcTopicTypes:1" Scheme="IptcTopicTypes"/>
    <FormalName Scheme="companycode">BRS</FormalName>
    <Description xml:lang="en">Bloomsbury Review Service</Description>
  </Topic>
</TopicSet>
...
<DescriptiveMetadata AssignedBy="BRS" Importance="normal" Confidence="High"
DateAndTime="20001231T1200+0000">
  ...
</DescriptiveMetadata>
...
</NewsML>

```

[NSK 解説]

メタデータがどのように割り当てられているのかを示すために、[assignment](#)エンティティが使われる。[assignment](#)エンティティは、[AssignedBy](#)属性、[Importance](#)属性、[Confidence](#)属性、[HowPresent](#)属性、[DateAndTime](#)属性からなる。

[assignment](#)エンティティが定義されている要素は、[Property](#)と[Origin](#)、それに、[RightsMetadata](#)と[DescriptiveMetadata](#)とその配下の一部の要素である。

[AssignedBy](#)属性はメタデータを割り当てている団体を表す。団体を示す文字列（例えば人の名前）を自由形式で記述するか、あるいは、その団体を識別するための[controlled vocabulary](#)を使って記述することもできる。

[Importance](#)属性はメタデータの重要性を示し、[Confidence](#)属性はメタデータの信用度を示す。[HowPresent](#)属性はメタデータを適用する方法を示している。これら3つの属性は対応する[controlled vocabulary](#)を使って記述されなければならない。

[DateAndTime](#)属性は、メタデータが割り当てられた日付を示す。オプションとして時間を記述することもできる。使用されるフォーマットは CCYYMMDDTHHMMSS±HHMM(世紀、年、月、日、時間区切り、時刻、分、秒、タイムゾーン区切り、時間、分)である。これは ISO 8601 で決められた標準フォーマットである。

これらの設定は、下位要素において明確に記述されていない場合、上位要素の記述内容が適用される。

5.5 NewsItem の構造

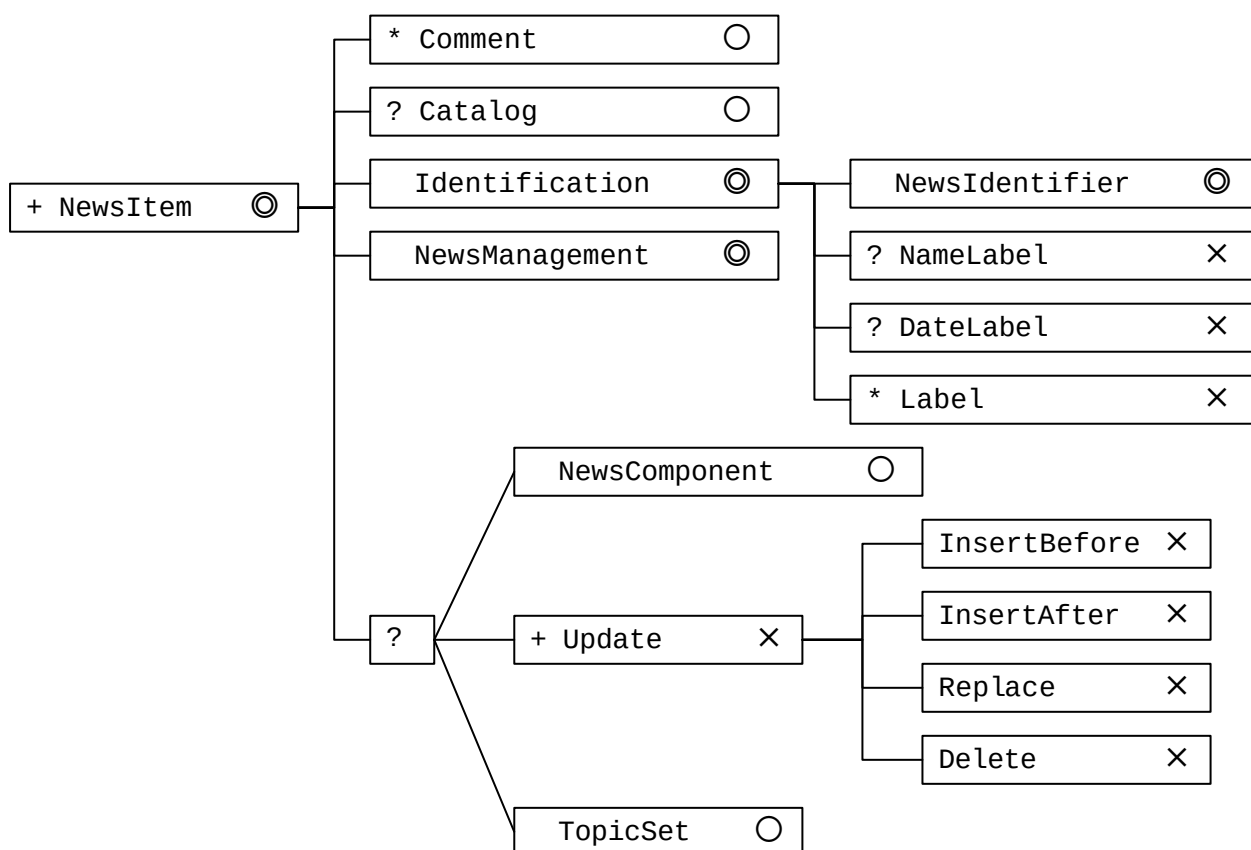
[NewsItem](#) は、イベントについて管理された情報の集合であり、いくつかのイベントに関するある時点での視点を表現する。[NewsItem](#) の [Identification](#) と [NewsManagement](#) 下位要素は、識別情報と容易な管理機能を提供する。さらに、[NewsItem](#) には [NewsComponent](#) か、同じ [NewsItem](#) の以前の版を修正する 1 つ以上の [Update](#) 要素、あるいは [TopicSet](#) を含むことができる。

[NewsItem](#) に適用できる [Catalog](#) は、[Catalog](#) 下位要素に含まれるか、オプションである [Catalog](#) 下位要素の [Href](#) 属性で参照される。[Href](#) 属性はこの文書あるいは、別の文書に存在する [Catalog](#) 要素へのポインタを示す。

```
<!ELEMENT NewsItem (Comment*, Catalog?, Identification, NewsManagement,
    ( NewsComponent | Update+ | TopicSet )? )>
<!ATTLIST NewsItem %localid;
    xml:lang CDATA #IMPLIED >

<!ELEMENT Identification (NewsIdentifier, NameLabel?, DateLabel?, Label*)>
<!ATTLIST Identification %localid; >
```

[ツリー図]



[NSK 解説]

[NewsItem](#) は NewsML 中のニュースを入れる実際のコンテナの役割を担っている。発行するニュースの管理単位である。

[NewsItem](#) は [Identification](#) という ID とバージョンの管理、[NewsManagement](#) というニュース管理機

構、コンテンツ自身やコンテンツ構造を表現する[NewsComponent](#)からなっている。

[NewsItem](#)でのコンテンツの管理単位はいくつかある。記事、写真など、それぞれのコンテンツを1つの[NewsItem](#)で管理する方法、また、複数のコンテンツを組み合わせて1つの[NewsItem](#)とする方法もある。

複数のコンテンツを組み合わせて表現する方法として、[NewsComponent](#)を階層化する方法、[ContentItem](#)を複数列挙する方法、また、他の[NewsItem](#)へのリンクを列挙して表現する[NewsItemRef](#)を利用する方法がある。

前述したように、[NewsItem](#)は[Identification](#)や[NewsManagement](#)によって ID や状態の管理をおこなうことができる。しかし、[NewsComponent](#)や[ContentItem](#)をこのように管理するための機構や規定はない。このため複数のコンテンツを組み合わせて管理する場合に、コンテンツ管理単位をよく吟味する必要がある。

他にも NewsML では、単純な複数コンテンツの表現方法として、NewsML 直下に[NewsItem](#)を複数並べて1つの NewsML として束ねる方法がある。しかし、単に複数の[NewsItem](#)を1つの NewsML にまとめただけであり、それぞれの[NewsItem](#)との関係や構造を表現するものではない。

[Comment](#)要素の値は人間が読むことを想定した形式で記述され、以下の多くの要素にオプションで記述することができる。

[AssociatedWith](#)要素、[ContentItem](#)要素、[Copyright](#)要素、[DerivedFrom](#)要素、[NewsComponent](#)要素、[NewsItem](#)要素、[NewsItemRef](#)要素、[Topic](#)要素、[TopicSet](#)要素、および[TopicSetRef](#)要素に対し、子要素に[Comment](#)要素を記述することで、その要素を補足説明する。

例えば[AssociatedWith](#)要素は、現在の[NewsItem](#)が[NewsItem](#)属性値で指定された[NewsItem](#)と関連があることを示すが、どのように関連するのかまではわからない。このとき、「これは前記事の続編です。」といった情報を[Comment](#)要素値に記述することにより、双方の関連性を説明できる。

また[NewsItemRef](#)要素では、[NewsItem](#)を含む理由を[Comment](#)要素で解説する。

[Contributor](#)要素、[Creator](#)要素、[Provider](#)要素、[SentFrom](#)要素、[SentTo](#)要素、および[Source](#)要素に対し、[Party](#)要素の直前に[Comment](#)要素を記述して、子要素の[Party](#)要素を解説する。

例えば、[Contributor](#)要素は生成されたニュースオブジェクトの修正や拡張を行った個人、会社、組織あるいはその組み合わせを[Party](#)要素で示すが、どのような修正をする担当者かまではわからない。そこで、[Comment](#)要素で「編集校閲者」や「引用者」といった記述を行う事で補足説明できる。

[Comment](#)要素はオプションとして [xml:lang](#)属性と [TranslationOf](#)属性を持つ。

[xml:lang](#)属性で[Comment](#)要素値の言語を指定することにより、さまざまな言語で記述することができる。

[TranslationOf](#)属性は IDREF 型で、同一文書内の同じ ID 型属性値をとる [Comment](#)要素を示すポイントとなる。

[NewsML 仕様の変更点：NewsML v1.1]

[NewsItem](#)の[Comment](#)要素に[FormalName](#)属性とその関連属性が、バージョン 1.1 で追加された。
([Comment](#)要素の変更後の定義、詳細は、「[5.4.2 SentFrom と SentTo](#)」を参照)

[NskNewsML:1 記述形式]

NSK ガイドラインではニュース素材の交換を目的とするため NewsML の [NewsItem](#) を一つとし、複数 [NewsItem](#) を並べることは禁止する。

NewsItem 要素	◎
NewsItem/Comment 要素	○
NewsItem/Catalog 要素	○
NewsItem/Identification 要素.....	◎
NewsItem/NewsManagement 要素	◎
NewsItem/NewsComponent 要素 ..	○
NewsItem/Update 要素	×
NewsItem/TopicSet 要素	○

[Comment](#)は変更箇所などのコメントを入れることができる。改版の情報に関しては「[5.11 NewsItemsの改版](#)」を参照。

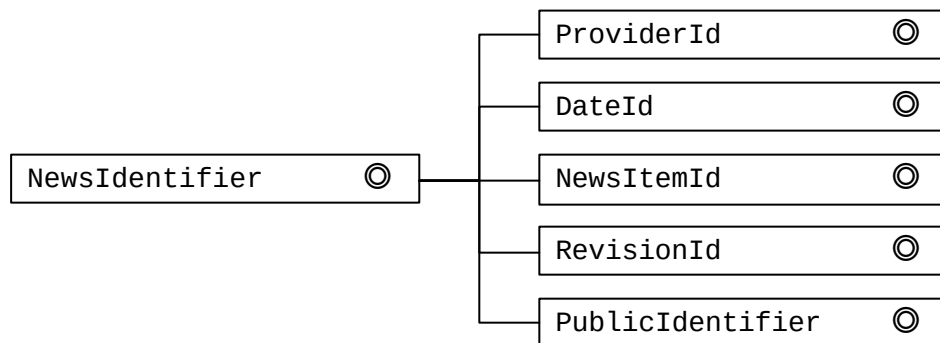
5.5.1 NewsItem の識別 ([NewsIdentifier](#))

ビジネスのワークフロー上での移動や、場所の移動、システム間での移動があっても、[NewsItem](#)を識別することができなければならない。それゆえに NewsML において、[NewsItem](#)は世界で唯一の識別子を[NewsIdentifier](#)要素で持つ必要がある。

[NewsIdentifier](#)は4つの下位要素[ProviderId](#)、[DateId](#)、[NewsItemId](#)、[RevisionId](#)と、これら4つの要素すべてを1本に結び付ける[PublicIdentifier](#)から成る。[NewsItem](#)における[NewsIdentifier](#)は、世界で唯一の識別子でなければならない。それゆえに、配信社は同じ[ProviderId](#)、[DateId](#)、[NewsItemId](#)、[RevisionId](#)を持つ2つの[NewsItem](#)がないことを保証しなければならない。もしも内容を変えたあとで[NewsItem](#)を再生成したら、それがわずかな変更であっても新たに[RevisionId](#)を新しい版に割り当てなければならない。

```
<!ELEMENT NewsIdentifier (ProviderId , DateId , NewsItemId , RevisionId, PublicIdentifier)>
```

[ツリー図]



[NSK 解説]

[NewsIdentifier](#)要素は[NewsItem](#)の識別子を表す。それは[NewsItem](#)がシステム間で移動やコピーを繰り返しても、その[NewsItem](#)を識別できるような世界で唯一のものでなければならない。同じ[NewsIdentifier](#)を持つ[NewsItem](#)があれば、それはまったく同じものである。配信社は同じ[NewsIdentifier](#)を持つ異なる[NewsItem](#)がないことを保証しなければならない。

[NewsIdentifier](#)要素は配信社を識別する[ProviderId](#)要素、日付を表す[DateId](#)要素、[NewsItem](#)の識別子となる[NewsItemId](#)要素、[NewsItem](#)の版を表す[RevisionId](#)要素と、これら4つの要素の内容を1つにつなげた[PublicIdentifier](#)要素から成る。

[NewsItem](#)の内容を変更した場合[RevisionId](#)に新しい版を割り当てる。

(例) [NewsIdentifier](#)の一般的な使用方法

```
<NewsIdentifier>
  <ProviderId>pressnet.or.jp</ProviderId>
  <DateId>20010426</DateId>
  <NewsItemId>ID0001</NewsItemId>
  <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
  <PublicIdentifier>urn:newsml:pressnet.or.jp:20010426:ID0001:1</PublicIdentifier>
</NewsIdentifier>
```

[NskNewsML:1 記述形式]

NewsIdentifier 要素 ◎
NewsIdentifier / ProviderId 要素 ◎
NewsIdentifier / DateId 要素 ◎
NewsIdentifier / NewsItemId 要素 ◎
NewsIdentifier / RevisionId 要素 ◎
NewsIdentifier / PublicIdentifier 要素 .. ◎

(例) NewsIdentifier の一般的な使用方法

```
<NewsIdentifier>  
  <ProviderId>pressnet.or.jp</ProviderId>  
  <DateId>20010426</DateId>  
  <NewsItemId>ID0001</NewsItemId>  
  <RevisionId PreviousRevision="0" Update="N">1</RevisionId>  
  <PublicIdentifier>urn:newsml:pressnet.or.jp:20010426:ID0001:1</PublicIdentifier>  
</NewsIdentifier>
```

5.5.1.1 ProviderId

[ProviderId](#)要素の内容は、[DateId](#)要素で識別される日付に配信社が保持しているドメイン名か、[Vocabulary](#)属性で分類された[URN](#)で識別される controlled vocabularies から書き出された配信社名でなければならない。[NewsIdentifier](#)が完全であれば正確に配信社を識別できる。

```
<!ELEMENT ProviderId (#PCDATA)>
<!ATTLIST ProviderId Vocabulary CDATA #IMPLIED >
```

次の例では、配信社はInternational Press Telecommunications Councilで、その[ProviderId](#)は[DateId](#)によって識別される日に、その配信社が保持しているドメイン名である。

```
<ProviderId>iptc.org</ProviderId>
<DateId>20001005</DateId>
```

[NSK 解説]

[ProviderId](#)要素は[NewsItem](#)を発行した配信社を識別する。それは[DateId](#)で示される日付において、その配信社が保持している正式なドメイン名でなければならない。[IPTC](#)の機能仕様書には2種類の方法が記述されているが、現状では[IPTC](#)自身がドメイン名の使用を推奨している。

[NskNewsML:1 記述形式]

[ProviderId](#) 要素◎

[ProviderId](#)/@[Vocabulary](#) ×

ドメイン名を使うこととする。

(例) [ProviderId](#)の一般的な使用例

```
<Provider Id>pressnet.or.jp</Provider Id>
```

5.5.1.2 DateId

[DateId](#)は ISO8601 の標準フォーマット(CCYYMMDD)の日付で、CCYY は 4 桁の年、MM は 2 桁の月、DD は 2 桁の日である。[DateId](#)は[NewsItem](#)を形式的に識別する部分なので、同じ[NewsItem](#)の連続したレビジョンを通して同じでなければならない。これは、現在のレビジョンを公開した日付を表したものではない。

```
<!ELEMENT DateId (#PCDATA )>
```

次の例では、2000年10月6日という日付は、その[NewsItem](#)が最初に作られた日でも、そうでなくてもよい。必要条件是、[ProviderId](#)がドメイン名ならば、その日付は配信社がそのドメイン名を保持している日の日付でなければならないということ、そのDateIdはこの[NewsItem](#)のすべての改版を通じて変えないということだけである。

```
<DateId>20001006</DateId>
```

[NSK 解説]

[DateId](#)は[NewsItem](#)を識別するための日付を表す。この要素は日単位で[NewsItemId](#)を唯一にするため

の日付である。

[DateId](#)は[NewsItem](#)の識別子の一部なので、[NewsItem](#)の内容を変更し改版しても前のものと同じでなければならない。

[NskNewsML:1 記述形式]

DateId 要素 ..◎

記述形式は ISO8601 の標準フォーマット(CCYYMMDD)で、CCYY は 4 桁の年、MM は 2 桁の月、DD は 2 桁の日である。

(例) DateId の一般的な使用例

```
<DateId>20010426</DateId>
```

5.5.1.3 NewsItemId

[NewsItemId](#)は、[NewsItem](#)の識別子である。同じ配信社から発する[NewsItem](#)の間で、[NewsItemId](#)と[DateId](#)の組み合わせは、唯一でなければならない。この条件を満たせば、[NewsItemId](#)は配信社が望む通りの形式をとることができる。[NewsItem](#)の名前は人が見て意味のある形式をとることができるが、必要条件ではない。

配信社は、オプションとして[NewsItemId](#)の値を[Vocabulary](#)属性に呼び出される[controlled vocabulary](#)に関連づけてもよい。[Vocabulary](#)属性の値は、httpのURL形式、またはNewsMLのURN形式、または#記号の後にその文書中にある[TopicSet](#)要素の[Duid](#)属性の値を続ける形をとる。[Scheme](#)属性が存在すれば、それは[controlled vocabulary](#)においてとり得る複数の[naming scheme](#)のうち、[NewsItemId](#)を制御するものと区別するのに役立つ。

```
<!ELEMENT NewsItemId (#PCDATA)>
<!ATTLIST NewsItemId Vocabulary CDATA #IMPLIED
          Scheme CDATA #IMPLIED >
```

```
<NewsItemId>IPTC approves NewsML 1.0</NewsItemId>
```

[NSK 解説]

[NewsItemId](#)は[NewsItem](#)の識別子である。それは同じ[ProviderId](#)で識別される配信社が発行した[NewsItem](#)の中で[DateId](#)が示す日付ごとに唯一なものでなければならない。

NewsItemId には自由形式で記述できるが、PublicIdentifier は使える文字に制限があることに注意すること（[5.5.1.5 PublicIdentifier](#)の項を参照）。

[NskNewsML:1 記述形式]

NewsItemId 要素 ...◎

(例) NewsItemId の一般的な使用例

```
<NewsItemId>ID0001</NewsItemId>
```

5.5.1.4 RevisionId

[RevisionId](#)は、これが[NewsItem](#)のどの版であるかについて示す正整数である。いかなる正整数でも使うことができるが、気をつけなければならないのは、同じ[ProviderId](#)、[DateId](#)、[NewsItemId](#)を持つ2つの[NewsItem](#)のうち、[RevisionId](#)値が大きい方が最新版でなければならないということである。「0」の[RevisionId](#)は許されない。また、[PreviousRevision](#)属性は存在しなければならない。その値は、[NewsItem](#)の前の版があればその[RevisionId](#)要素の値と等しく、前の版がなければ「0」でなければならない。[NewsItem](#)が1つあるいは複数の[Update](#)要素を含むならば、[Update](#)属性には「U」がセットされなければならない。[NewsItem](#)が[NewsManagement](#)データの置き換えセットだけからなるならば、[Update](#)属性は「A」がセットされなければならない。これらのどちらのケースでもない場合は、[Update](#)属性は「N」がセットされなければならない。

```
<!ELEMENT RevisionId (#PCDATA)>
<!ATTLIST RevisionId PreviousRevision CDATA # REQUIRED
                  Update CDATA # REQUIRED >
```

次の例では、現在の版番号は「1」であり、前の版はない。

```
<RevisionId PreviousRevision="0" Update="N">1</RevisionId>
```

次の例では、現在の版番号は「2」であり、前の版番号は「1」であった。

```
<RevisionId PreviousRevision="1" Update="N">2</RevisionId>
```

次の例では、[RevisionId](#)要素の[Update](#)属性の値が「U」であるため、[NewsItem](#)が1つあるいは複数の[Update](#)要素を含むことを示す。そして、それは前の版を修正するのに役立つ。現在の版番号は20001023であり、前の版番号は20001005であった。[PreviousRevision](#)の値が連続している必要がないことに注意すること。必要条件は同じ[NewsItem](#)の前のどの版の数字よりも値が大きくなければならないということだけである。

```
<RevisionId PreviousRevision="20001005" Update="U">20001023</RevisionId>
```

[NSK 解説]

[RevisionId](#)は[NewsItem](#)の版数を示す正の整数である。

[NewsItem](#)を改版する場合は、[ProviderId](#)、[DateId](#)、[NewsItemId](#)は元のままとし、[RevisionId](#)の値を以前のものより大きくする。同じ[ProviderId](#)、[DateId](#)、[NewsItemId](#)を持つ2つの[NewsItem](#)があるならば、[RevisionId](#)の大きい方が最新の版である。

[PreviousRevision](#)属性値は[NewsItem](#)の前の版（[RevisionId](#)の値）を表す。最初の版には「0」を設定する。

[Update](#)属性は通常、「N」を設定する。[NewsItem](#)の[NewsManagement](#)情報のみを変更する場合[Update](#)属性は「A」を設定し、[RevisionId](#)はもとのままにする。この場合、[NewsComponent](#)を付加してはいけない。

[NewsItem](#)に[Update](#)要素が含まれる場合「U」を設定する。

(例) [Update](#)属性が「N」の使用例

- ・ 初版（前の版がない）の場合

```
<RevisionId PreviousRevision="0" Update="N">1</RevisionId>
```

- ・ 現在の版が「2」で前の版が「1」の場合

```
<RevisionId PreviousRevision="1" Update="N">2</RevisionId>
```

(例) [Update](#)属性が「A」の使用例

以下の例はすでにある[NewsItem](#)をキャンセル (Canceled) することを表す。

※ このとき[PublicIdentifier](#)にAが付加されていることに注意すること。

※ [Status](#)の意味と記述形式は[5.6.4.Status](#)の項を参照。

```
<NewsML>
  . . .
  <NewsItem>
    <NewsIdentifier>
      . . .
      <RevisionId PreviousRevision="0" Update="A">1</RevisionId>
      <PublicIdentifier>urn:newsml:prennet.or.jp:20010426:ID0001:1A</PublicIdentifier>
    </NewsIdentifier>
    <NewsManagement>
      . . .
      <Status FormalName="Canceled"/>
    </NewsManagement>
  </NewsItem>
</NewsML>
```

[NskNewsML:1 記述形式]

RevisionId 要素 ◎

RevisionId/@PreviousRevision ・◎

RevisionId/@Update ◎

Update 属性値は「N」か「A」のみである。(「U」は[NskNewsML:1]では使用停止)

すでに発行されている NewsItem の NewsManagement の情報のみを変更する場合 (Status の変更など) に「A」を利用する。この場合は指定された版の NewsItem の NewsManagement 情報に直接反映される。

5.5.1.5 PublicIdentifier

XML 1.0 の仕様によって定義される意味で、[NewsIdentifier](#) 要素は、[NewsItem](#) のために [PublicIdentifier](#) を用意する。これは、[NewsItem](#) のために以下のような [URN](#) の形式をとる。

```
urn:newsml:{ProviderId}:{DateId}:{NewsItemId}:{RevisionId}{RevisionId@Update}
```

ここで{x}は「[NewsIdentifier](#)の x 下位要素の内容」を意味する。{x@y}は「[NewsIdentifier](#)の x 下位要素の y 属性の属性値を意味する。例外として、[RevisionId](#)要素の[Update](#)属性がNのデフォルト値を持つ場合、[URN](#)から省かれる。

[URN](#)の中に直接含むことができる文字セットが制限される点に注意すること。許される文字は、[IETF](#) (Internet Engineering Task Force) が指定したもので、その RFC (Request for Comments) 2141 番に示されている。この文書は、<http://www.ietf.org/rfc/rfc2141.txt> で入手できる。[URN](#)文字セットで認められていない文字は、RFC2141に記述された一連の正しい文字列に置き換えなければならない。

この[URN](#)の存在により[NewsItem](#)が、他のXML要素またはリソースからポインタによって明確に参照づけられることができる点に注意すること。そのようなポインタ内で[RevisionId](#)、その前の「:」記号、その次の[Update](#)属性の記述が省略された場合、ポインタは解析された最新の版を示す。

<ELEMENT PublicIdentifier (#PCDATA)>

[NewsIdentifier](#)の以下の例は、[RevisionId](#)要素の[Update](#)属性が値「N」を持つ場合に、[PublicIdentifier](#)がとる形式を示す。[NewsItem](#)の内容は1つの[NewsComponent](#)か1つの[TopicSet](#)で、[Update](#)の集合ではないことを示している。

```
<NewsIdentifier>
  <ProviderId>iptc.org</ProviderId>
  <DateId>20001006</DateId>
  <NewsItemId>NewsML Approved</NewsItemId>
  <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
  <PublicIdentifier>urn:newsml:iptc.org:20001006:NewsML%20Approved:1</PublicIdentifier>
</NewsIdentifier>
```

[URN](#)のスペース文字は、%記号にスペースのための16進の文字コード（20）を続けて表さなければならないので、[NewsItemId](#)要素の内容のスペースは、[PublicIdentifier](#)要素の内容において%20となることに注意する。

下記の例では、[RevisionId](#)要素の[Update](#)属性は値「U」を持ち、[NewsItem](#)の内容は1つ以上の[Update](#)の集合であることを示している。

```
<NewsIdentifier>
  <ProviderId>iptc.org</ProviderId>
  <DateId>20001006</DateId>
  <NewsItemId>i123</NewsItemId>
  <RevisionId PreviousRevision="20001005" Update="U">20001023</RevisionId>
  <PublicIdentifier>urn:newsml:iptc.org:20001006:i123:20001023U</PublicIdentifier>
</NewsIdentifier>
```

この例では、[RevisionId](#)と[PreviousRevision](#)の値が連続していないが、[RevisionId](#)の値が[PreviousRevision](#)の値よりも大きい点に注意すること。ニュース配信社が1から始まる連続した数字よりむしろ改訂値を産み出すために、日付を使う方を選んだように見える。これは完全に認められる方法である。

この[NewsItem](#)を受け取る際、[Update](#)によって示された変化を反映する完全な[NewsItem](#)を生成するために、システムは[Update](#)指示を[NewsItem](#)の前の版に適用しなければならない。この結果、[NewsItem](#)は以下の[NewsIdentifier](#)を持つことになる。そこにおいて、[RevisionId](#)要素の[Update](#)属性は値「N」を持ち、アップデートを示す文字は[PublicIdentifier](#)の文字列の終わりから省かれる。

```
<NewsIdentifier>
  <ProviderId>iptc.org</ProviderId>
  <DateId>20001006</DateId>
  <NewsItemId>i123</NewsItemId>
  <RevisionId PreviousRevision="20001005" Update="N">20001023</RevisionId>
  <PublicIdentifier>urn:newsml:iptc.org:20001006:i123:20001023</PublicIdentifier>
</NewsIdentifier>
```

最後に、[RevisionID](#) を全く指定しない [URN](#) のポインタが、参照が解析されたときに、[NewsItem](#)の最新版が何であろうともそれを指し示すことに注意すること。それ故、urn:newsml:iptc.org:20001006:i123 の文字列は、今の例で[NewsItem](#)の現在の最新版が何であれそれを示す。

[NSK 解説]

[PublicIdentifier](#) 要素は[NewsItem](#)の公開識別子を表し、他の NewsML から参照する際に使用される重

要な要素である。

これは以下のようなURN形式でProviderId、DateId、NewsItemId、RevisionId、RevisionIdのUpdate属性値をつなぎ合わせたものである。

urn:newsml:{ProviderId}:{DateId}:{NewsItemId}:{RevisionId}{RevisionId@Update}

RevisionId要素のUpdate属性が「N」の場合、URNから省かれることに注意しなければならない。

この NewsML URN により他の NewsML 文書からNewsItemを参照することが可能である。その際RevisionIdの直前の「:」記号以降がすべて省略された場合には、参照されるNewsItemの最新版を示すことになる。

※ NewsML URN 補足

NewsML URN の NID (ネームスペース識別子、ここではnewsml) は IANA に認められ、RFC2611 で公開されている。そして URN は RFC3085 ([ftp://ftp.rfc-editor.org/in-notes/rfc3085.txt](http://ftp.rfc-editor.org/in-notes/rfc3085.txt)) に示されている。

(例) NewsML URN を利用したNewsItemの参照例

・版を指定して参照する例

<DerivedFrom NewsItem="urn:newsml:pressnet.or.jp:20010426:ID0001:1">

...

</DerivedFrom>

※DerivedFromの意味と記述形式はDerivedFromの項を参照。

・最新版を参照する例

<AssociatedWith NewsItem="urn:newsml:pressnet.or.jp:20010426:ID0001">

...

</AssociatedWith>

※AssociatedWithの意味と記述形式はAssociatedWithの項を参照。

※ NewsML URN に使える文字の補足

URN の中に含むことのできる文字には制限がある。それはIETFによって RFC2141 (<http://www.ietf.org/rfc/rfc2141.txt>) で示されている。

使用できる文字には以下の文字がある。

・A-Z、a-z、0-9、

・"(", ")", "+", ",", "-", ".", ":", "=", "@", ";", "\$", "_", "!", "*", ""

これ以外の文字を使用する場合、UTF-8エンコーディングされた1～6バイトの文字を%記号のあとに16進文字(0-9、A-F)を続けて表す。例えば、スペース文字は「%20」、%記号自体は「%25」として表す。

(例) NewsItemIdにURN で使用できない文字が含まれる場合の記述例

(スペース文字が含まれる場合)

<NewsIdentifier>

```
<ProviderId>pressnet.or.jp</ProviderId>
<DateId>20010426</DateId>
<NewsItemId>NSK NewsML Level1</NewsItemId>
<RevisionId PreviousRevision="0" Update="N">1</RevisionId>
<PublicIdentifier>urn:newsml:pressnet.or.jp:20010426:NSK%20NewsML%20Level1:1</Public
Identifier>
</NewsIdentifier>
```

[NskNewsML:1 記述形式]

PublicIdentifier 要素 ••◎

(例) PublicIdentifier の標準的な記述例

```
<NewsIdentifier>
  <ProviderId>pressnet.or.jp</ProviderId>
  <DateId>20010426</DateId>
  <NewsItemId>ID0001</NewsItemId>
  <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
  <PublicIdentifier>urn:newsml:pressnet.or.jp:20010426:ID0001:1</PublicIdentifier>
</NewsIdentifier>
```

5.5.2 非形式識別子

上で述べた形式的な識別メカニズムに加えて、NewsML は[NewsItem](#)を人的に識別するための一連の[Label](#)要素を提供する。NewsML システムに関する限り、これらは任意のストリングであって、強力な識別メカニズムの提供として期待はできない。それらの唯一の目的は、非形式の交換および通信に、あるいはユーザー・インターフェースの一部として人間が特別の[NewsItem](#)を識別できる便利な方法を提供することである。

5.5.2.1 NameLabel

[NameLabel](#) 要素は、[NewsItem](#)の識別を支援するための名前として、人間が使用するストリングを含んでいる。その形式は、配信社によって決定される。例えば、それは[SlugLine](#)要素の原文と同じであっても、システムはslugLineとして[NameLabel](#)を処理してはならない。それは、人間による[NewsItem](#)の識別を支援することができるという事実を越えて、[NameLabel](#)内のストリングの性質に関して何も想定することができないためである。

```
<!ELEMENT NameLabel (#PCDATA )>
<!ATTLIST NameLabel %localid; >
```

```
<NameLabel>IPTC approves NewsML 1.0</NameLabel>
```

[NSK解説]

[NameLabel](#)要素の設定値は、文字データを表す「#PCDATA」である。また[NameLabel](#)要素は属性値としてlocalidエンティティ、すなわち[Duid](#)属性と[Euid](#)属性を持つことができる。

[NameLabel](#)要素は[Identification](#)の子要素で出現回数は、0回または1回で、出現順序は、[NewsIdentifier](#), [NameLabel](#), [DateLabel](#), [Label](#)の順番である。

5.5.2.2 DateLabel

[DateLabel](#)要素は、日付のストリング表現を含む。ラベルの目的はユーザーへの利便性なので、ISO 標準の日付形式には必ずしも従わなくてよい。

```
<!ELEMENT DateLabel (#PCDATA )>
<!ATTLIST DateLabel %localid; >
```

```
<DateLabel>6 October 2000</NameLabel>
```

[NSK解説]

[DateLabel](#)要素の設定値は、文字データを表す「#PCDATA」である。また[DateLabel](#)要素は属性値としてlocalidエンティティ、すなわち[Duid](#)属性と[Euid](#)属性を持つことができる。

[DateLabel](#)要素は[Identification](#)の子要素で出現回数は、0回または1回で、出現順序は、[NewsIdentifier](#), [NameLabel](#), [DateLabel](#), [Label](#)の順番である。

5.5.2.3 Label

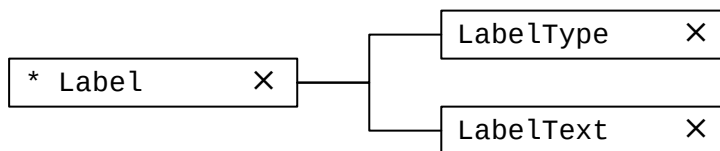
[Label](#)要素は、[LabelType](#)と[LabelText](#)の下位要素からなり、[NewsItem](#)のための、オプションの読みやすいラベルである。[LabelText](#)は、特定の[LabelType](#)の[Label](#)を構成するテキストである。[LabelType](#)はユーザーが定義するタイプのラベルである。[FormalName](#)属性の値はラベル・タイプのための形式的な名前である。その意味および設定値は、[Vocabulary](#)と[Scheme](#)属性によって識別された[controlled vocabulary](#)により決定される。

```
<!ELEMENT Label (LabelType, LabelText)>
<!ATTLIST Label %localid; >

<!ELEMENT LabelType EMPTY>
<!ATTLIST LabelType %localid;
                %formalname; >

<!ELEMENT LabelText (#PCDATA)>
<!ATTLIST LabelText %localid; >
```

[ツリー図]



```
<Label>
  <LabelType FormalName="ShortRef"
  Vocabulary="urn:newsml:mydomain.com:20001006:MyLabelTypes:1" Scheme="labeltypes"/>
  <LabelText>NewsMLv1.0</LabelText>
</Label>
```

[NSK解説]

[Label](#)要素は、[LabelType](#)要素と[LabelText](#)要素を下位要素として持ち、出現回数は1回、順序は、[LabelType](#)、[LabelText](#)。[LabelType](#)は要素値をもたず、属性値としてlocalidエンティティとformalnameエンティティを持つ。そのため属性値として[Duid](#)属性と[Euid](#)属性、[FormalName](#)属性、[Vocabulary](#)属性、[Scheme](#)属性を持つことができる。

[LabelText](#)は文字データを表す「#PCDATA」で、属性値としてlocalidエンティティ、すなわち[Duid](#)属性と[Euid](#)属性を持つ。

[Label](#)要素は[Identification](#)の下位要素で、出現回数は0回以上の繰り返し、出現順序は、[NewsIdentifier](#)、[NameLabel](#)、[DateLabel](#)、[Label](#)の順番である。

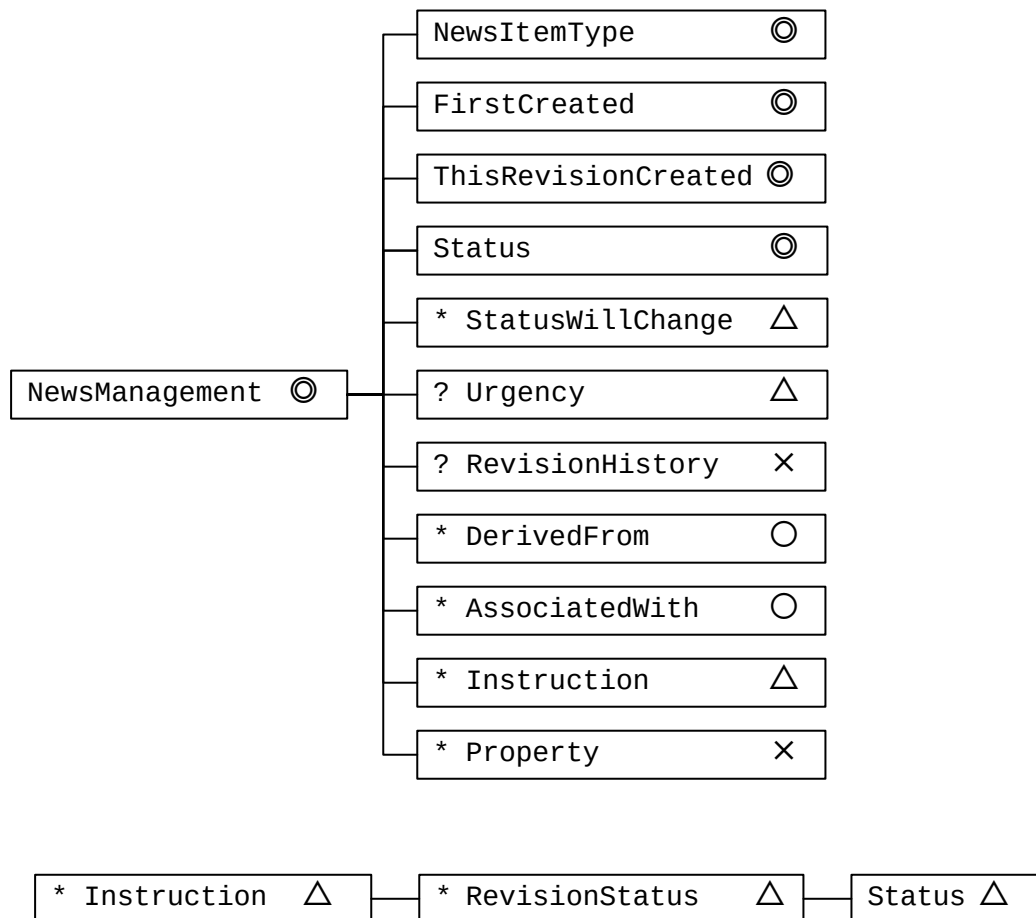
[Label](#)要素は[Identification](#)の下位要素である為、Revisionの変更を伴わずに変更することができる。

5.6 NewsManagement

[NewsManagement](#)要素は、[NewsItem](#)のタイプ、情報、履歴、ステータス、他の[NewsItem](#)との関係、適用されるべき特別の指示や追加のプロパティなど[NewsItem](#)の管理に関連する情報を与える。

```
<!ELEMENT NewsManagement (NewsItemType , FirstCreated , ThisRevisionCreated ,  
Status , StatusWillChange* , Urgency? , RevisionHistory? , DerivedFrom* ,  
AssociatedWith* , Instruction* , Property* )>  
<!ATTLIST NewsManagement %localid; >
```

[ツリー図]



[NSK解説]

[NewsManagement](#)要素は、[NewsItem](#)ごとの管理情報を表す。管理情報はニュースやデータなどであるかを表す種別 ([NewsItemType](#))、最初に作成された日時 ([FirstCreated](#))、Revisionが作成された日時 ([ThisRevisionCreated](#))、利用可否を示すステータス ([Status](#))、ステータス変化の事前通知 ([StatusWillChange](#))、緊急度 ([Urgency](#))、改訂履歴 ([RevisionHistory](#))、派生元 [NewsItem](#) ([DerivedFrom](#))、関連する [NewsItem](#) ([AssociatedWith](#))、受け取る側への指示 ([Instruction](#))、付加的な管理データ ([Property](#)) などを表す情報を含む。

[NewsItem/Identification/NewsIdentifier/RevisionId](#) の update 属性値が「A」のときは、[NewsManagement](#)内の情報の書き換えを意味することに注意。（「Updateの方法」参照）

(例)

```
<NewsManagement>
  <NewsItemType FormalName="News" />
  <FirstCreated>20010424T150100+0900</FirstCreated>
  <ThisRevisionCreated>20010424T210100+0900</ThisRevisionCreated>
  <Status FormalName="Usable" />
  <Urgency FormalName="5" />
  <DerivedFrom NewsItem="urn:newsml:provider.or.jp:20010312:0001:2">
    <Comment>第2報</Comment>
  </DerivedFrom>
  <AssociatedWith NewsItem="urn:newsml:provider.or.jp:20010313:P0001:1">
    <Comment>記事併用写真</Comment>
  </AssociatedWith>
</NewsManagement>
```

[NskNewsML:1記述形式]

NewsManagement要素		◎
NewsItemType要素		◎
FirstCreated要素		◎
ThisRevisionCreated要素		◎
Status要素		◎
DerivedFrom要素		○
AssociatedWith要素		○
RevisionHistory要素		×
Property要素は		×

5.6.1 NewsItemType

NewsItemType 要素は、[NewsItem](#) のタイプを指定する。[FormalName](#) 属性の値は、ニュース-アイテム・タイプのための形式的な名前である。その意味と許される値は、[Vocabulary](#) 属性と [Scheme](#) 属性によって識別される [controlled vocabulary](#) により決定される。

```
<!ELEMENT NewsItemType EMPTY >
<!ATTLIST NewsItemType %localid;
                %formalname; >
```

```
<NewsItemType FormalName="News"
Vocabulary="urn:newsml:iptc.org:20001006:IptcNewsItemTypes:1"
Scheme="IptcNewsItemTypes"/>
```

[NSK 解説]

NewsItemType 要素は、[NewsItem](#) の種別を示す。

NewsItemType の値とその意味

TopicSet 名：topicset.iptc-newsitemtype-ja.xml (RevisionId:4)

NewsItemType の値 (FormalName)	意味 (Description)
News	記事、写真、ビデオ、音声レポートなど、ニュースイベントの基本的な報道内容
Data	通常、編集者が編集したり、修正したりすることのない非叙述情報、あるいは配信社によって第三者からユーザーへ機械的に流される情報。たとえば、スポーツの試合結果や株価など。
Advisory	編集連絡や配信予定などのニュースイベントに関連した外部には出さない情報
Maintenance	NewsML を送受信するシステムの通信経路（チャンネル）に関するシステムからシステムへのメッセージ
TopicSet	NewsML で外部参照する、コントロールド・ボキャブラリ（TopicSet ファイル）を単独で送る場合に使用する。
DTD (IPTC のみが利用する)	XML ドキュメントの DTD (文書型定義)を送る場合に使用する。
Document (IPTC のみが利用する)	ニュースではない文書
Catalog	NewsML で外部参照する、Resource エlement と TopicUse エlement の入った Catalog ファイルを単独で送る場合に使用する。
Alert	重要な突発的なニュースについての事前の警告メッセージ。
NewsManagementMode	ニュースプロバイダーによって使用されるニュース管理のモード情報を規定する非公開コミュニケーション。
Schema	XML 文書の正規構造を明示する XML スキーマ。

(例) <NewsItemType FormalName="News"/>

上記例は当該 NewsItemType 要素の [NewsItem](#) の型 ([FormalName](#) 属性値) は "News" であることを示している。

[NskNewsML:1 記述形式]

NewsItemType 要素 …… ◎

5.6.2 FirstCreated

この必須の要素は[NewsItem](#)が最初に作られた日付と、オプションとして時間を、ISO 8601 標準フォーマットで示す。

```
<!ELEMENT FirstCreated (#PCDATA)>
<!ATTLIST FirstCreated %localid; >
```

この例は[NewsItem](#)が最初に作られた日が、ローカル時間で2000年10月6日14時で、協定世界時(UTC)より2時間進んでいることを示している。

```
<FirstCreated>20001006T1400+0200</FirstCreated>
```

[NSK 解説]

[FirstCreated](#)とは、この[NewsItem](#)の最初の Revision ([5.5.1.4RevisionId](#)参照)が作成された日付情報を示す要素である。

年月日は必須だが、時間はオプションである。

NewsML で時間を表す要素は、[DateAndTime](#)要素、[FirstCreated](#)要素、[ThisRevisionCreated](#)要素がある。これらの要素の値はすべて ISO 8601 標準フォーマットとして、年月日は必須だが、時間はオプションとなっている。例えば、これらの要素の値を統一して時間、分、秒までを記述すれば、初版の作成日時、改版の作成日時、これらの版を配信した日時が秒単位まで管理、区別することが可能となる。

[NskNewsML:1 記述形式]

FirstCreated 要素 .. ◎

5.6.3 ThisRevisionCreated

この必須の要素は[NewsItem](#)が改訂された日付と、オプションとして時間を、ISO 8601 標準フォーマットで示す。

```
<!ELEMENT ThisRevisionCreated (#PCDATA)>
<!ATTLIST ThisRevisionCreated %localid; >
```

下の例は、[NewsItem](#)のこの版がローカル時間で2000年10月6日16時15分に作られ、協定世界時(UTC)より2時間進んでいることを示している。

```
<ThisRevisionCreated>20001006T1615+0200</ThisRevisionCreated>
```

[NSK 解説]

[ThisRevisionCreated](#)とは、この Revision に[NewsItem](#)が改訂された日付情報を示す要素である。年月日は必須だが、時間はオプションである。オプションに関しては、[5.6.2FirstCreated](#)を参照。

[NskNewsML:1 記述形式]

ThisRevisionCreated 要素 ・ ◎

5.6.4 Status

この必須の要素は、[NewsItem](#)の現在のステータスを示す。[FormalName](#)属性の値は、ステータスのための形式的な名前である。その意味および許可される値は、[Vocabulary](#)属性と[Scheme](#)属性によって識別される[controlled vocabulary](#)によって決定される。

```
<!ELEMENT Status EMPTY >
<!--ATTLIST Status %localid;
        %formalname; -->
```

```
<Status Vocabulary="urn:newsml:iptc.org:20001006:IptcStatus:1" Scheme="IptcStatus"
FormalName="Embargoed"/>
```

[NSK解説]

[Status](#)要素は[NewsItem](#)が利用可能かどうかを示し、送信側の指示により[Status](#)要素の値を変更することができる。

IPTCボキャブラリの値と意味は以下の通り。

1) Usable (公開可能)

この版の[NewsItem](#)を利用して良い。

2) Canceled (公開取消)

既に送られたこの版の[NewsItem](#)を公開取り消しする場合に使用する。

この場合、既に送った[NewsItem](#)の[RevisionId](#)の値が同じで[Update](#)属性が“A”とした上で、[Status](#)値を“Canceled”とする。

“Canceled”の場合、[NewsComponent](#)は付けない ([Update](#)の項参照)。

3) Embargoed (公開待機)

この版の[NewsItem](#)は公開待ちとする。

この値は、[StatusWillChange](#)と併せて使用されることが考えられる。公開待機期間(後日、公開予定)のニュースは、[Status](#)を“Embargoed”とし、[StatusWillChange](#)で公開日時を指定する使い方がある。

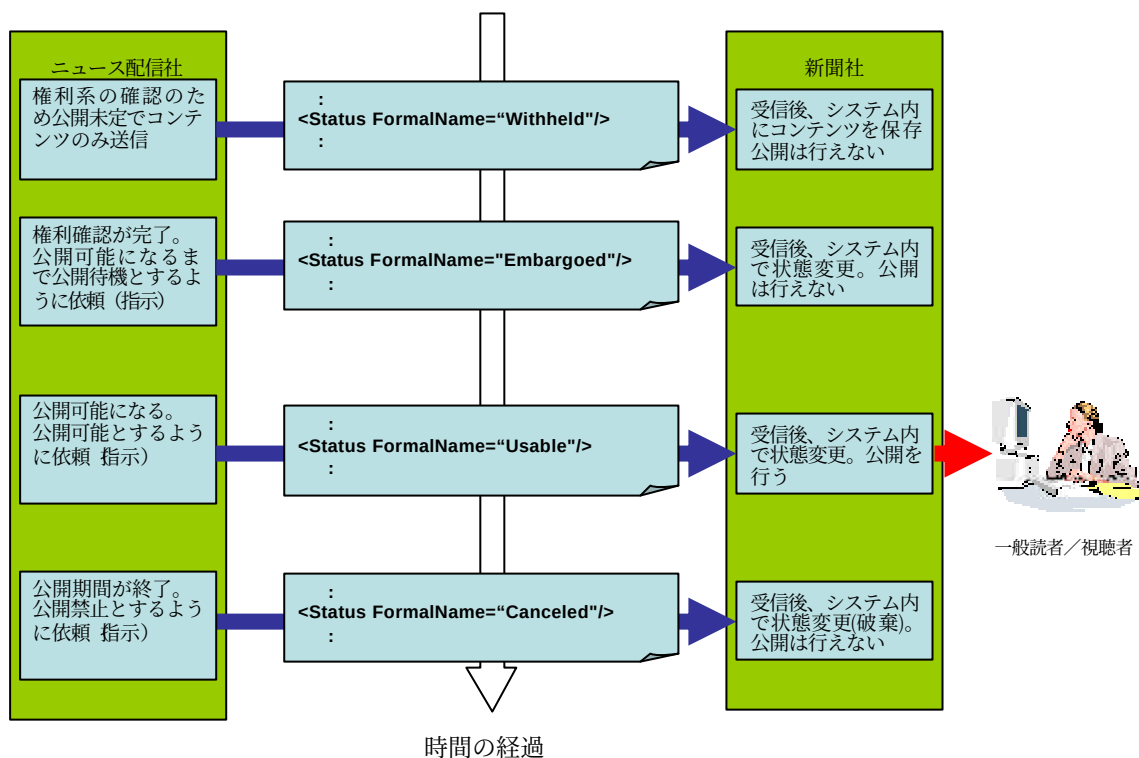
公開の予定 (ステータス変化する予定) が立っている場合に利用される。

4) Withheld (公開未定)

この版の[NewsItem](#)の公開は未定である。

[NewsItem](#) (の情報)が不正確であるときや、法的問題となる可能性がある場合に利用される。“Withheld”の[NewsItem](#)を受けても、(送信元から) 何らかのフォローがあるとは限らない。

(例) あるニュースコンテンツの状態遷移



上の図のように配信済みの`NewsItem`に対して受信側は送信側から指示された`Status`要素の値の変更を行う。但し、`Status`要素の値の最終状態は`Canceled`なので、`Canceled`の状態から他の値に変更することはできない。

配信済みの`NewsItem`の状態を変更する方法については「[Updateの方法](#)」参照。

[NskNewsML:1記述形式]

`Status`要素 … ◎

ボキャブラリはIPTCのボキャブラリを使用する。

5.6.5 StatusWillChange

オプションの [StatusWillChange](#) 要素は、指定された日時に自動的に起こるステータス変化の事前通知を行う。[StatusWillChange](#) 中の必須の [FutureStatus](#) 要素は、[NewsItem](#) が将来の指定日時に持つステータスを示す。[FormalName](#) 属性の値は、ステータスのための形式的な名前である。その意味と許される値は、[Vocabulary](#) 属性と [Scheme](#) 属性によって識別される [controlled vocabulary](#) によって決定される。必須の [DateAndTime](#) 要素は、ISO 8601 標準フォーマットを用い、ステータス変化が起きる日時を示す。例えば「embargoed (公開待機)」[Status](#) の [NewsItem](#) は、[StatusWillChange](#) 要素を持つことにより指定された時間に「usable (公開可能)」になる。これは公開待機が終わり、[NewsItem](#) が公開される時間を、事前にアナウンスすることに相当する。この要素の複数使用は、[NewsItem](#) の事前計画された変化を前もって表すことができる。

```
<!ELEMENT StatusWillChange (FutureStatus , DateAndTime )>
<!ATTLIST StatusWillChange %localid; >

<!ELEMENT FutureStatus EMPTY >
<!ATTLIST FutureStatus %localid;
                %formalname; >
```

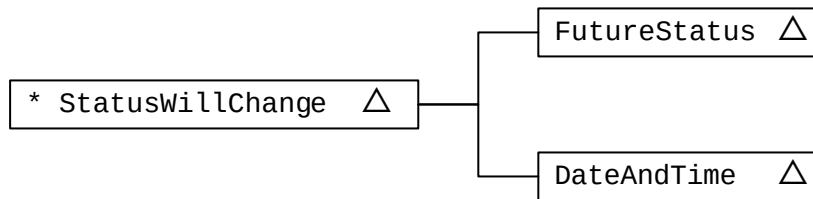
以下の例では、[NewsItem](#) がその生成時には公開待機 (embargoed) だったのが、[UTC](#) 時間の2000年7月7日12時に公開可能(usable)になることを示している。[NewsItem](#) のステータスの変更は、ニュース配信社内のローカルなイベントではなく、グローバルなイベントであることに留意すること。なぜならばその[NewsItem](#) はグローバルな識別子を持ち、そのステータスは世界中で適応されるからである。

```
<Catalog>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcStatus:1</Urn>
    <DefaultVocabularyFor Scheme="IptcStatus" Context="Status | FutureStatus"/>
  </Resource>
</Catalog>
...
<Status FormalName="Embargoed"/>
<StatusWillChange>
  <FutureStatus FormalName="Usable"/>
  <DateAndTime>20000707T1200+0000</DateAndTime>
</StatusWillChange>
```

2つの[DefaultVocabularyFor](#)要素が、[XPath](#)の文法を代替パターンでマッチングするのに使用することにより、1つに結合されていることに注目すること。以下の例において、[DefaultVocabularyFor](#)要素で表現されているのは、IPTC status vocabularyを「要素名=Statusまたは要素名=[FutureStatus](#)」のパターンにマッチするあらゆるデータに適応するということである。

```
<Resource>
  <Urn>urn:newsml:iptc.org:20001006:IptcStatus:1</Urn>
  <DefaultVocabularyFor Scheme="IptcStatus" Context="Status | FutureStatus"/>
</Resource>
```

[ツリー図]



[NSK解説]

[StatusWillChange](#)要素は指定日時に指定ステータスに変更することを示す。

例えば、公開禁止期間(後日、公開予定)のニュースを先に提供する場合に、公開日時をこの要素で指定する。

ステータス変更時刻以降、該当する版の NewsItem を urn など呼び出した場合、ステータスの変更が反映された NewsItem を示す。

また、[StatusWillChange](#)要素は、複数回記述することができる。これにより、[Status](#)要素の値を変更させるために[NewsManagement](#)を送信しなくても、事前に[StatusWillChange](#)要素によって[Status](#)要素の値の変更を指示する事が可能である。つまり、あらかじめ[Status](#)要素の値の変更時間が分かっている場合は、送信側からの指示がなくてもFutureStatus要素と[DateAndTime](#)要素を使い、受信側で[Status](#)要素の値を変更することができる。但し、FutureStatus要素の値と[DateAndTime](#)の値は未来に対して矛盾の無いように進むよう注意すること。[Status](#)要素の値に変更を与えるFutureStatus要素の有効な変化過程は、次のものが一般的に考えられる。

[FutureStatus](#)のボキャブラリは[Status](#)と同じく [IPTC](#)のボキャブラリを使用する。また、[DateAndTime](#)は、[5.4.3 DateAndTime](#)に準じる。

▽目的の時間に公開したい場合

- Embargoed([Status](#)要素の値)? Usable(FutureStatus 要素の値)

▽目的の時間に削除したい場合

- Usable([Status](#)要素の値)? Canceled(FutureStatus 要素の値)

また、[StatusWillChange](#)が複数記述可能になったため、上記を組み合わせる次のような表現が可能となった。

▽目的の時間に公開したのち、一定時間後に削除したい場合

- Embargoed([Status](#)要素の値)? Usable(FutureStatus 要素の値) ? Canceled(FutureStatus 要素の値)

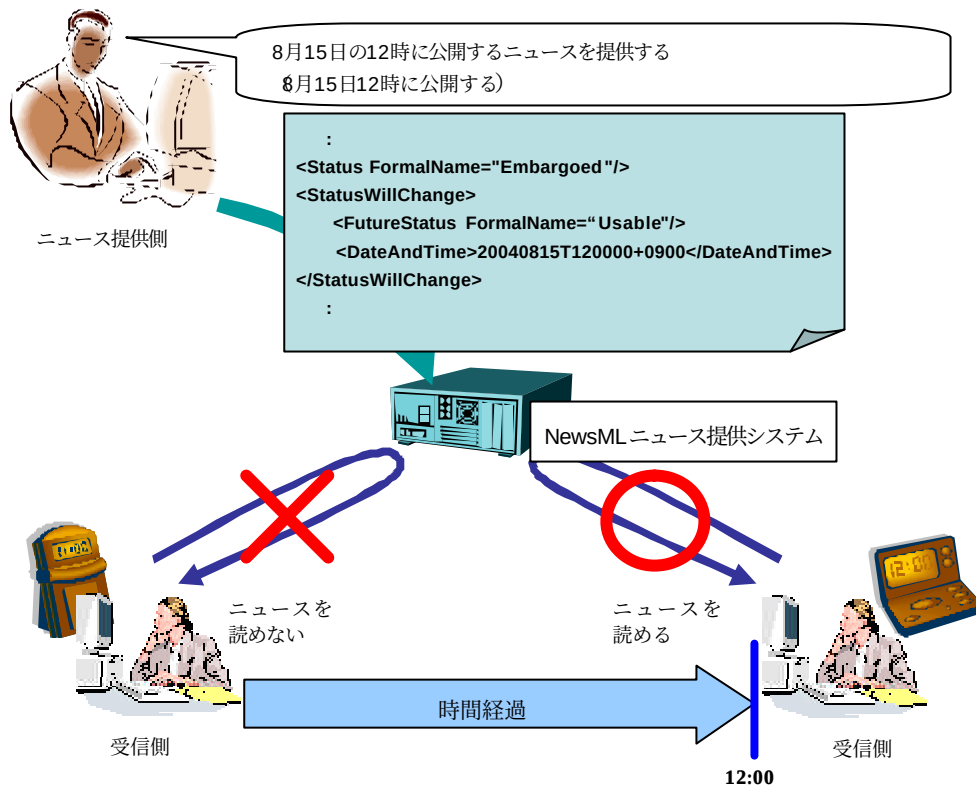
一度 Canceled になった[Status](#)要素の値は、他の値に変更することはできないので注意すること。

[StatusWillChange](#)要素を使用する場合の値の組み合わせ一覧

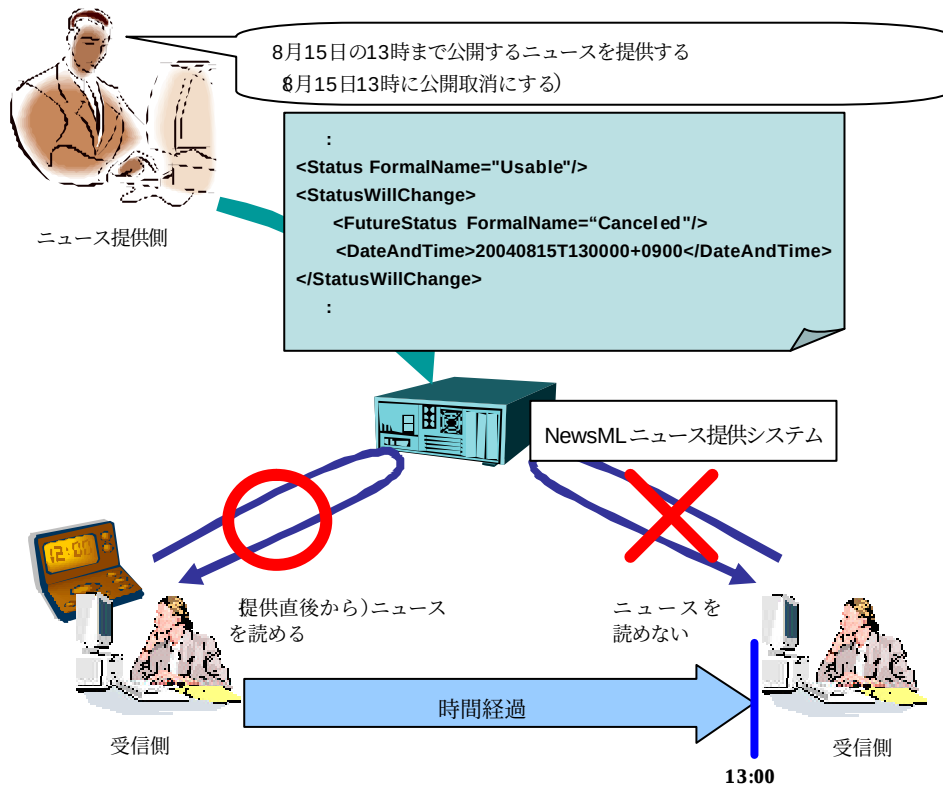
Status の値	Usable	Withheld	Embargoed		Canceled
FutureStatus の値	Canceled	Canceled	Usable	Canceled	StatusWillChange を記述できない
			Canceled		

なお、NskNewsML:1.2 では StatusWillChange 要素は 0 回、または 1 回。繰り返しは認めない。

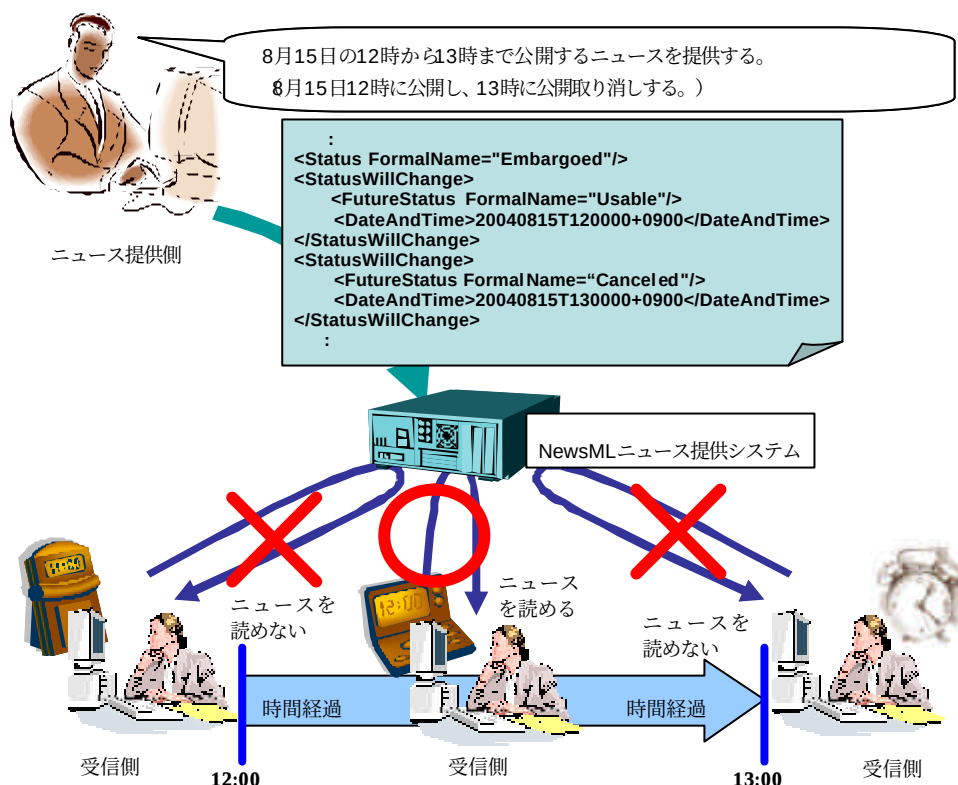
(例1) Embargoed→Usable の変化過程



(例2) Usable→Canceled の変化過程



(例3) Embargoed→Usable→Canceled の変化過程



[NewsML 仕様の変更点：NewsML v1.1]

[StatusWillChange](#)は、Version1.0では0回または1回しか記述できなかった。しかし、Version1.1になって、複数回の記述ができるようになった。

[NskNewsML:1記述形式]

StatusWillChange 要素…△

[NskNewsML:1.2記述形式]

StatusWillChange 要素 △

NewsML1.1でStatusWillChange要素は複数回記述可能となったが、NskNewsML:1.2では0回または1回。繰り返しは認めない。

5.6.6 Urgency

オプションの [Urgency](#) 要素は `NewsItem` の緊急性を示す。 [FormalName](#) 属性の値は緊急度を示す形式名である。その意味と許される値は、 [Vocabulary](#) 属性及び [Scheme](#) 属性によって識別される [controlled vocabulary](#) によって決定される。

```
<!ELEMENT Urgency EMPTY>
<!--ATTLIST Urgency %localid;
      %formalname; -->
```

```
<Urgency Vocabulary="urn:newsml:iptc.org:20001006:IptcUrgency:1" Scheme="IptcUrgency"
FormalName="1"/>
```

[NSK解説]

[Urgency](#) 要素は、`NewsItem` に対する編集上の重要性を伴う緊急度を示す。

値は”1”～”9”までの整数をセットする。最も緊急度が高い場合は”1”をセットし、最も緊急度が低い場合は”8”をセットする。通常の緊急度の場合は”5”をセットする。”9”は特別な値でユーザが独自に定義することができる。（この場合、あらかじめ配信先と、値の定義について取り決めをしておく必要がある）。 `Importance` 属性は、その属性を含むメタデータの重要度を示す。

5.6.7 RevisionHistory

オプション（0回もしくは1回出現）の [RevisionHistory](#) 要素は、`NewsItem` の改訂履歴（revision history）を含むファイルへのポインタを示す。このポインタは [Href](#) 属性によって示される。配信社はこのファイルをどのような文章や構造でも使用して好きなように表現すればよい。

```
<!ELEMENT RevisionHistory EMPTY>
<!--ATTLIST RevisionHistory %localid;
      Href CDATA #REQUIRED -->
```

この例では、`NewsItem` の改訂履歴についての情報は、`NewsItem` 自体を入れたディレクトリの `history` サブディレクトリ内の `rev_1376.log` ファイルにあることが示されている。

```
<RevisionHistory Href="../history/rev_1376.log"/>
```

[NSK 解説]

[RevisionHistory](#) 要素は、`NewsItem` の改訂履歴を記載したファイルへのポインタを示す。このファイルへのポインタは [Href](#) 属性によって示される。配信社はこのファイルをどのような文章や構造で使用しても良い。

5.6.8 DerivedFrom

オプションで反復可能（0回以上出現）の [DerivedFrom](#) 要素はこれがどのNewsItemから派生したかのポインタを示す。NewsItem属性によって関連するNewsItemを示し、属性値はhttp URLか、NewsML URNが使用できる。その由来を示すのにオプションで [Comment](#) が使える。ポインタの型は、[DerivedFrom](#) 要素の [FormalName](#) 属性で任意に指定される。[DerivedFrom](#) 要素の [Vocabulary](#) 属性は、その [FormalName](#) の意味を定義する [controlled vocabulary](#) へのポインタである。もし [Scheme](#) 属性があれば語彙中のネーミングスキームを識別し、この形式名に適用される。

この例では、現在のNewsItem がどのNewsItem から派生したかを、URNによって示している。[Comment](#) 要素は、もとのNewsItemとの従属関係を示すために使われている。ニュース配信社が元のNewsItemとの[DerivedFrom](#) 関係を使用して新たにNewsItem を作るか、または同じNewsItemの新しい版を作るかは、独自の判断、手順に基づいてかまわない。[DerivedFrom](#) が採用されるのは、NewsItemが修正され、異なったニュースサービスで公開されるときであり、同一のニュースサービスで公開される場合は新しい版が使われてもよい。NewsMLはこれについては特別な運用手順を示すものではない。

```
<DerivedFrom NewsItem="urn:newsml:iptc.org:20001006:NewsML%201.0%20approved" >
<Comment>Statement from the Chair of the NewsML Steering Committee.</Comment>
</DerivedFrom>
```

[ツリー図]



[NSK解説]

[DerivedFrom](#) 要素では、当該の NewsItem が、他の NewsItem から“派生した”ことが表現できる。また、オプションの NewsItem 属性で派生元の NewsItem を示すことができる。更に NewsMLv1.1 から追加された [FormalName](#) 属性を利用して、派生の種別を明確に区別することができる。[FormalName](#) 属性にセットされる [controlled vocabulary](#) について明確な指針はないが、[Comment](#) 要素との使い分けは、[Comment](#) 要素の値は編集者がセットする自由記述情報なのに対して、[FormalName](#) 属性は、システムに対して明確に指示する値となる。NewsML では、NewsItem の修正が行われた場合は、新しい版として配信する場合と新たなニュースとして配信することができる。新たなニュースとして配信する場合は、

```
<!ELEMENT DerivedFrom (Comment*)>
<!ATTLIST DerivedFrom %localid;
                NewsItem CDATA #IMPLIED FormalName CDATA #IMPLIED
                Vocabulary CDATA #IMPLIED Scheme CDATA #IMPLIED >
```

[DerivedFrom](#) 要素を使用して派生元との関連性を示すことができる。

[NewsML 仕様の変更点：NewsML v1.1]

オプションの [FormalName](#) 属性、[Vocabulary](#) 属性、[Scheme](#) 属性が追加になった。

[Comment](#)要素は派生元NewsItemの位置づけを記述する。

次の例は、派生元のNewsItemNewsML URNで指定している。[Comment](#)要素により、派生元のNewsItemが第2報であることを示している。

```
<DerivedFrom NewsItem="urn:newsml:provider.or.jp:20010312:0001:2" >
  <Comment>第2報</Comment>
</DerivedFrom>
```

[NskNewsML:1記述形式]

DerivedFrom要素○
Comment要素○

[NskNewsML:1.2記述形式]

DerivedFrom要素	○
DerivedFrom/@FormalName	×
DerivedFrom/@Vocabulary	×
DerivedFrom/@Scheme	×
Comment 要素	○

NewsML1.1でFormalName属性・Vocabulary属性・Scheme属性が追加となったが、使用停止とする。

5.6.9 AssociatedWith

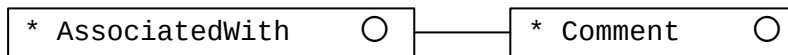
オプションで反復可能（0回以上出現）の[AssociatedWith](#)要素は関連のある[NewsItem](#)へのポインタを示す（例えば、その[NewsItem](#)を含む関連記事、または写真集の一部分など）。[NewsItem](#)属性によって[NewsItem](#)の参照先を示し、属性値はURIか、NewsML URNが使用できる。オプションの[Comment](#)は関連性についての説明を示すのに使用できる。ポインタの型は、[AssociatedWith](#)要素の[FormalName](#)属性で任意に指定される。[AssociatedWith](#)要素の[Vocabulary](#)属性は、その[FormalName](#)の意味を定義する[controlled vocabulary](#)へのポインタである。もしScheme属性があれば語彙中のネーミングスキームを識別し、この形式名に適用される。

```
<!ELEMENT AssociatedWith (Comment*)>
<!ATTLIST AssociatedWith %localid;
  NewsItem CDATA #IMPLIED
  FormalName CDATA #IMPLIED
  Vocabulary CDATA #IMPLIED
  Scheme CDATA #IMPLIED>
```

この例では、現在の[NewsItem](#)が、URNによって識別されたものと関連があるということを示している。[Comment](#)要素はその関連性を示すために使われる。

```
<AssociatedWith NewsItem="urn:newsml:iptc.org:20001006:NewsML%201.0%20approved" >
  <Comment>This is a sequel to the previous story.</Comment>
</AssociatedWith>
```

[ツリー図]



[NSK解説]

[AssociatedWith](#)要素では、当該の NewsItem が、他の NewsItem と“関連している”ことが表現できる。また、オプションの NewsItem 属性により関連先の NewsItem を示すことができる。更に NewsMLv1.1 から追加された [FormalName](#) 属性を利用して、関連の種別を明確に区別することができる。[FormalName](#) 属性にセットされる [controlled vocabulary](#) について明確な指針はないが、“photo”、“article”などをセットした場合は、関連先の NewsItem を解析せずに、何であるかの情報がある程度持たせることも可能である。[Comment](#) 要素との使い分けについては、[Comment](#) 要素の値は編集者がセットする自由記述情報なのに対して、[FormalName](#) 属性は、システムに対して明確に指示する値となる。NewsItem 同士のリンク情報は、[AssociatedWith](#) 要素と [DerivedFrom](#) 要素の2種類のみなので、リンク情報を補完するために [FormalName](#) 属性や [Comment](#) 要素を使用することができる。

この例では、この NewsItem が、別の NewsItem と関連があるということを示している。関連があるのがどの NewsItem なのかを識別するために、PublicIdentifier を NewsML urn として使っている。

[Comment](#) 要素（省略可）でこの NewsItem と別の NewsItem がどのように関連しているかを記述している。

```

<AssociatedWith NewsItem="urn:newsml:provider.or.jp:20010313:P0001:1">
  <Comment>記事併用写真</Comment>
</AssociatedWith>
  
```

[NewsML 仕様の変更点：NewsML v1.1]

オプションの [FormalName](#) 属性、[Vocabulary](#) 属性、[Scheme](#) 属性が追加になった。

[NskNewsML:1 記述形式]

AssociatedWith要素 ○
AssociatedWith/Comment要素 ○

[NskNewsML:1.2記述形式]

AssociatedWith要素 ○
AssociatedWith/@FormalName ×
AssociatedWith /@Vocabulary ×
AssociatedWith /@Scheme ×
AssociatedWith/Comment 要素 ○

NewsML1.1 で [FormalName](#) 属性・[Vocabulary](#) 属性・[Scheme](#) 属性が追加となったが、使用停止とする。

5.6.10 Instruction

オプションで反復可能（0回以上出現）の [Instruction](#) 要素は、ニュース配信社からNewsItemを受け取る側へのinstruction（指示）を示す。[Instruction](#)の特別なケースとしては、最新の版のNewsItemが、受信側のシステム内に残っているNewsItemの前の版のステータスに与える影響を示す。この場合 [Instruction](#) 要素は1つ以上の[RevisionStatus](#)要素を持つ。あるいは、[FormalName](#)属性の属性値はその指示の形式名である。この形式名の意味と使用可能な値は[controlled vocabulary](#)によって示される。[controlled vocabulary](#)は[Vocabulary](#)属性及び[Scheme](#)属性によって指定される。

[RevisionStatus](#)要素は、現在の版が公開された結果、前の改訂が今どういうステータスにあるかを示す。オプションのRevision属性は整数値であり、問題となる改訂の[RevisionId](#)と同じにする。これが無い場合、前の改訂すべてに例外なくここでのステータスが適用される。

```
<!ELEMENT Instruction (RevisionStatus*)>
<!ATTLIST Instruction %localid;
                    %formalname; >

<!ELEMENT RevisionStatus (Status)>
<!ATTLIST RevisionStatus %localid;
                    Revision CDATA #IMPLIED >
```

この例では、NewsItemの前の版のすべてのステータスが「公開取消」になることを示す。

```
<Instruction FormalName="CancelAll" Vocabulary="#MyInstructionCodes">
  <RevisionStatus>
    <Status FormalName="Canceled"/>
  </RevisionStatus>
</Instruction>
```

この例では、第1版と第2版のステータスは「公開取消」だが、第3版はまだ「公開可能」であることを示す。

```
<Instruction FormalName="MostRecentStillUsable" Vocabulary="#MyInstructionCodes">
  <RevisionStatus Revision="1">
    <Status FormalName="Canceled"/>
  </RevisionStatus>
  <RevisionStatus Revision="2">
    <Status FormalName="Canceled"/>
  </RevisionStatus>
  <RevisionStatus Revision="3">
    <Status FormalName="Usable"/>
  </RevisionStatus>
</Instruction>
```

[NSK解説]

[Instruction](#)要素は、ニュース配信社から受信社に対しての指示を出す場合に使用される。

特に過去に複数の版を送信している場合に、それぞれの版のステータス（公開可、不可など）を変更することができる。例えば、第1版と第2版を「公開取消」とし、第3版のみ「公開可能」とする場合に利用される。この場合、対象となる版をRevision属性で示す。

```
<Instruction FormalName="MostRecentStillUsable" Vocabulary="#MyInstructionCodes">
  <RevisionStatus Revision="1">
    <Status FormalName="Canceled"/>
```

```
</RevisionSatus>
<RevisionStatus Revision="2">
  <Status FormalName="Canceled"/>
</RevisionSatus>
<RevisionStatus Revision="3">
  <Status FormalName="Usable"/>
</RevisionSatus>
</Instruction>
```

過去に送信したすべての版に対して、まとめて指示を出すことも可能で、その場合には、RevisionStatus要素にはRevision属性を付けずに送ればよい。

次の例では、過去に送信したすべての版が「公開取消」となることを示す。

```
<Instruction FormalName="CancelAll" Vocabulary="#MyInstructionCodes">
<RevisionStatus>
<Status FormalName="Canceled"/>
</RevisionSatus>
</Instruction>
```

5.6.11 Property

[Property](#) 要素は[ContentItem](#)、[Topic](#)、[NewsComponent](#)、[NewsItem](#)のいくつかのプロパティの値を表すのに使われる。そのプロパティには形式名が付いていなければならない、複雑なプロパティを扱うための下位プロパティを含んでもよい。

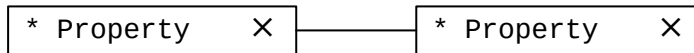
[Property](#)は名称を持ち、単純な値または、さらに入り組んだプロパティのセットでなる複雑な値のどちらかである。[Value](#)属性は[Property](#)の値を表す文字列である。[ValueRef](#)属性は[Property](#)の値へのポインタを与えている。これは[TopicSet](#)にある[Topic](#)かもしれないし、何か他のデータの一部であるかもしれない。もし[Value](#)属性と[ValueRef](#)属性の両方が与えられたなら、[ValueRef](#)は[Property](#)の実際の値を表し、[Value](#)は文字列の表現か記憶を与えるだけのものである。AllowedScheme属性がもしあれば、propertyの[Value](#)属性の内容に結びついた[Scheme](#)を表している。[AllowedValues](#)属性がある場合、それは一連のプロパティの値を定めた[controlled vocabulary](#)へのポインタである。この場合http URLかNewsML URN、または#文字の後に現在の文章中にある要素の[Duid](#)が続く[fragment identifier](#)であろう。ポインタは外部の[controlled vocabulary](#)を示す[Resource](#)要素か、それ自体が[controlled vocabulary](#)である [TopicSet](#)要素のどちらかを必ず参照しなければならない。

```
<!ELEMENT Property (Property*)>
<!ATTLIST Property %localid;
    %formalname;
    %assignment;
    Value CDATA #IMPLIED
    ValueRef CDATA #IMPLIED
    AllowedScheme CDATA #IMPLIED
    AllowedValues CDATA #IMPLIED >
```

以下の例は、[Characteristics](#)要素配下にある[Property](#)の形式名のための[default vocabulary](#)が[Characteristics](#)であることを、Catalogが宣言しているもので、ボキャブラリはwww.mydomain.comのvocabsというサブディレクトリの中で見つけることができる。[Context](#)属性の値は2つの//文字を含むXPath文法の形式で、[Characteristics](#)内にある[Property](#)の入れ子の度合いを任意に示すものである。Width(幅) [Property](#)は、Quantity(量) [Property](#)とUnit(単位) [Property](#)を含んでいる。3つの名前(Width、Quantity、Unit)は上記で宣言した[controlled vocabulary](#)によってすべて管理される。Quantityの値は7.5で、Unitの値はURNがurn:newsml:mydomain.com:20010101:Units:1のリソース内にある要素である。このURN文字列の後ろに続く#cmは、cmという値である[Duid](#)属性を持つ要素を結び付ける[fragment identifier](#)である。DuidはNewsML DTDでID属性となるように宣言されており、これはどのように[fragment identifier](#)がXML文書内で解析するかということである。この例では、URNが[TopicSet](#)を識別するだろう。また、[fragment identifier](#)は、“センチメートル(centimeter)”単位であることを示す[Description](#)下位要素を持つ[Topic](#)へ行き着くだろう。例えば、[Topic](#)はまた、メートル単位の長さのISO標準記述を指す[Href](#)属性を持つかもしれない。

```
<Catalog>
  <Resource Duid="resource1">
    <Urn>urn:newsml:mydomain.com:20010101:Characteristics:3</Urn>
    <Url>www.mydomain.com/vocabs/characteristics.xml</Url>
    <DefaultVocabularyFor Context="//Property"/>
  </Resource>
</Catalog>
...
<Characteristics>
  <Property FormalName="Width">
    <Property FormalName="Quantity" Value="7.5"/>
    <Property FormalName="Unit" ValueRef="urn:newsml:mydomain.com:20010101:Units:1#cm"/>
  </Property>
</Characteristics>
```

[ツリー図]



[NSK 解説]

[Property](#)要素は NewsML に規定されている要素では表現できない情報を記述する。基本的な使い方としては[FormalName](#)属性で種別を表し、[Value](#)属性にその値を記述する。AllowedScheme 属性を使って、[Value](#)属性の内容に結びつく [Scheme](#)を指定できる。AllowedScheme 属性は NskNewsML レベル 1.2 から追加された属性で、他の属性と同様 Property そのものが使用停止の個所は、下位の AllowedScheme も使用停止とする。

[Property](#)要素の中に[Property](#)要素を入れ子構造で記述する場合は、親の[Property](#)要素は[FormalName](#)属性だけを記述する。その値は、子の[Property](#)に記述する。[Value](#)属性値は原則として自由形式である。[AllowedValues](#)属性には、[Value](#)属性値の候補の入っている[TopicSet](#)ファイルへのポインタを記述する。

(例) 入れ子構造のない場合

```

<Property FormalName="ISBN" Value="ISBN-87966-870-2" />
<Property FormalName="PageCount" Value="462" />
  
```

(例) 入れ子構造がある場合

```

<Property FormalName="NskOriginatedArea">
  <Property FormalName="NskCountry" Value="USA"/>
  <Property FormalName="NskLocation" Value="ハワイ島ホノルル沖"/>
</Property>
  
```

[NewsML 仕様の変更点：NewsML v1.1]

AllowedScheme 属性が追加された。

[NskNewsML:1 記述形式]

Metadata/Property要素・・・	○
Characteristics/Property要素	△
その他のPropertyは・・・・・	×

コラム：独自Metadataの追加について

以下、[Property](#)を使ったメタデータの拡張方法について記述する。

[Property](#)要素は入れ子にすることで階層表現が可能であるが、構造を定義できないため、複雑な構造にすることは避けるべきである。

[Property](#)要素は[FormalName](#)属性で種別、[Value](#)属性でその値を記述する。

[Value](#)属性のとり得る値は自由記述形式、または Catalog/Resource でボキャブラリを宣言することにより[controlled vocabulary](#)として使用できる。

階層構造をとった場合は最下層の[Property](#)要素のみ[Value](#)をもつこととする。

1 階層の場合も含めて、最下層の[Property](#)要素に[Value](#)がない場合はエラーとする。

例)

- 1 階層並列表記の場合

```
<Property FormalName="ImageWidth" Value="1000"/>
```

```
<Property FormalName="ImageHigh" Value="1500"/>
```

- 2 階層表記の場合

```
<Property FormalName="NskOriginatedArea">
```

```
  <Property FormalName="NskCountry" Value="USA"/>
```

```
  <Property FormalName="NskLocation" Value="ハワイ島ホノルル沖"/>
```

```
</Property>
```

各社独自に[Metadata](#)を使用する場合は、独自の[MetadataType](#)のボキャブラリを用意できる。[MetadataType](#)要素の[FormalName](#)属性の値により拡張されたメタデータのタイプを記述する。NSK、[IPTC](#)のボキャブラリを拡張することにより NSK、[IPTC](#)で定義する[FormalName](#)（メタデータタイプ）を同時に使用することもできる。

[Property](#)のボキャブラリについての規定は無い。[IPTC](#)、NSK、その他独自のものが使用できる。

例) 独自 [Metadata](#) で写真の値段を入れた場合

1. Valueが自由記述形式の場合

```
<Metadata>
```

```
  <Catalog>
```

```
    <Resource>
```

```
      <!--MetadataType のボキャブラリを宣言（MyMetadataType 中では、MediaPriceが定義されている） -->
```

```
      <Urn>urn:newsml:Mycompany.or.jp:20010602:MyMetadataType:1</Urn>
```

```
      <DefaultVocabularyFor Context="MetadataType/@FormalName"/>
```

```
    </Resource>
```

```
    <Resource>
```

```
      <!--Property のボキャブラリを宣言（MyMediaPrice 中では、PhotoPriceが定義されている） -->
```

```
-->
```

```
      <Urn>urn:newsml:Mycompany.or.jp:20010602:MyMediaPrice:1</Urn>
```

```
      <DefaultVocabularyFor Context="Property/@FormalName"/>
```

```
    </Resource>
```

```
  </Catalog>
```

```
  <MetadataType FormalName="MediaPrice"/>
```

```
  <Property FormalName="PhotoPrice" Value="5000"/>
```

```
</Metadata>
```

2. Value に controlled vocabulary の値を使用する場合。

```

<Metadata>
  <Catalog>
    <Resource>
      <!--MetadataType のボキャブラリを宣言（ MyMetadataType 中では、MediaPriceが定義され
ている） -->
      <Urn>urn:newsml:Mycompany.or.jp:20010602:MyMetadataType:1</Urn>
      <DefaultVocabularyFor Context="MetadataType/@FormalName"/>
    </Resource>
    <Resource>
      <!-- Property のボキャブラリを宣言（ MyMediaPrice 中で PhotoPriceが定義されている） -->
      <Urn>urn:newsml:Mycompany.or.jp:20010602:MyMediaPrice:1</Urn>
      <DefaultVocabularyFor Context="Property/@FormalName"/>
    </Resource>
    <Resource>
      <!-- Value のボキャブラリを宣言（ MyPhotoPrice 中で Price1 が定義されている） -->
      <Urn>urn:newsml:Mycompany.or.jp:20010602:MyPhotoPrice:1</Urn>
      <DefaultVocabularyFor Context="Property[@FormalName='PhotoPrice']/@Value"/>
    </Resource>
  </Catalog>
  <MetadataType FormalName="MediaPrice"/>
  <Property FormalName="PhotoPrice" Value="Price1"/>
</Metadata>

```

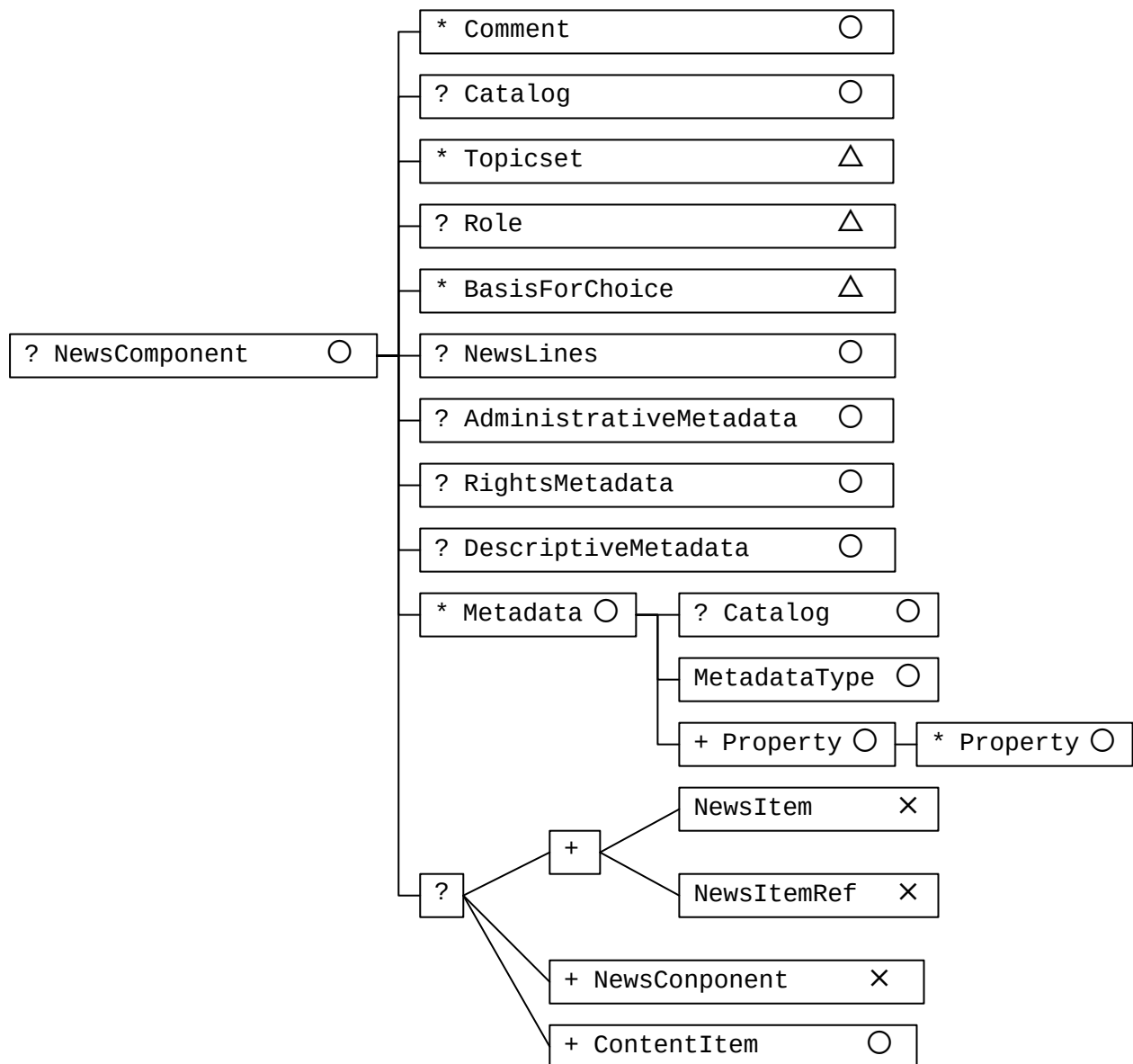

5.7 NewsComponent の構造

ニュースに特徴的なこととして、複数のデータオブジェクト、たとえば文章、写真とそのキャプション、グラフィックスが同時に流れることが多い。その上、複数の完結した記事を一緒に流し、たとえば週の主な記事の要約のように、それらを関連付けて扱うことや、特定のイベントやテーマに関係する記事の検索結果として扱うことがしばしば必要となる。この複雑さを解決するために、NewsComponentsを使って管理する。それらは[news objects](#)同士の関係構造を明確にしている。

[NewsComponent](#)は[news objects](#)の入れ物となる。これは[news objects](#)のお互いの関係を示すことや、メタデータを[NewsComponent](#)に関連付けることに使われる。[Essential](#)属性は、この[NewsComponent](#)がその属性を含む[NewsComponent](#)の意味にとって必須であると配信社がみなしているかどうかを示す。[EquivalentsList](#)属性は、この中に含まれるNewsItem か [NewsItemRef](#)、[NewsComponent](#) か [ContentItem](#)が、お互いに内容か意味（または両方）において同等であるかどうかを示している。[NewsComponent](#)の[Role](#)下位要素は、それを含む[NewsComponent](#)の中にある[NewsComponent](#)が担う役割を明らかにしている。NewsItemの一番外側の[NewsComponent](#)に[Role](#)属性の値を明記する必要はない。[FormalName](#)属性の値は、[Role](#)の形式名である。その意味と取りうる値は[controlled vocabulary](#)によって決まる。

```
<!ELEMENT NewsComponent (Comment* , Catalog? , TopicSet* , Role? , BasisForChoice* ,  
NewsLines? , AdministrativeMetadata? , RightsMetadata? , DescriptiveMetadata? , Metadata* , ((NewsItem |  
NewsItemRef)+ | NewsComponent+ | ContentItem+)?)>  
<!ATTLIST NewsComponent %localid;  
          Essential (yes | no ) "no"  
          EquivalentsList (yes | no ) "no"  
          xml:lang CDATA #IMPLIED >  
  
<!ELEMENT Role EMPTY>  
<!ATTLIST Role %localid;  
          %formalname; >
```

[ツリー図]



[NSK 解説]

[NewsComponent](#)の役割は、コンテンツの構造化のための入れ物とそのメタデータの関係付けである。

[NewsComponent](#)の構造化に関しては、ContentItem、NewsComponent、NewsItem(NewsItemRef)のいずれかを[NewsComponent](#)で包含する方法がある。通常は実際のコンテンツを格納した[ContentItem](#)をもつが、コンテンツをさらに階層化する場合には[NewsComponent](#)を入れ子にしたり、NewsML 自体（あるいはそのリンク）を包含したりできる。

この構造化機構を利用することで複雑な構造をもつニュースの表現が行える。しかし、そのデータを扱えるようにするためには、かなり複雑な仕組みや運用が必要になるため、実装においてはある程度の制限やルールを規定したほうがよい。

複数のコンテンツの関係や役割を示すための機構として、[Essential](#)、[EquivalentsList](#)、[Role](#)、[BasisForChoice](#)がある。ここでは簡単な説明だけ記載しておく、詳しくはそれぞれの説明を参照のこと。

[Essential](#)は、複数の[NewsComponent](#)がある場合に、どの[NewsComponent](#)が必須かどうかを示す Yes/No を指定する。省略した場合は'no'となる。

[EquivalentsList](#)は、[NewsComponent](#)内にある複数のコンテンツが、それぞれ同等なデータの場合 Yes となる。同等なデータとは、記事であれば内容が同じで[XML](#)表記、html 表記、Text 表記というデータが 列挙してある場合、画像であれば同じ図柄で解像度の違うデータが列挙してある場合などが当てはまる。 省略した場合は'no'となる。

[Role](#)は、複数ある[NewsComponent](#)の役割を[FormalName](#)にて記載し、それぞれを区別するために用い る。

[BasisForChoice](#)は[NewsComponent](#)内に複数列挙したコンテンツを識別するためにどの情報を用いる かを示すようになっている。

[Role](#)要素は[FormalName](#)属性、[Vocabulary](#)属性、[Scheme](#)属性があり、[FormalName](#)属性に許可された 値は、[Vocabulary](#)属性と[Scheme](#)属性によって識別される controlled vocabularies により決定される。

[NskNewsML:1 記述形式]

NewsComponent 要素……………○

NewsComponent/@xml:lang……………○

xml:lang 属性値は ISO639 に準じる。日本語を使うときは"ja"を記述する。指定がない場合は親要素の 属性値に従う。

(例) <NewsComponent xml:lang="ja">

NewsComponent/Comment 要素・○

注意喚起、変更個所などの連絡事項を入れる。

(例) <Comment>愛知県注意。とりあえず送ります。過去形手直しあり。</Comment>

NewsComponent/Catalog 要素……○

Role 要素……………△

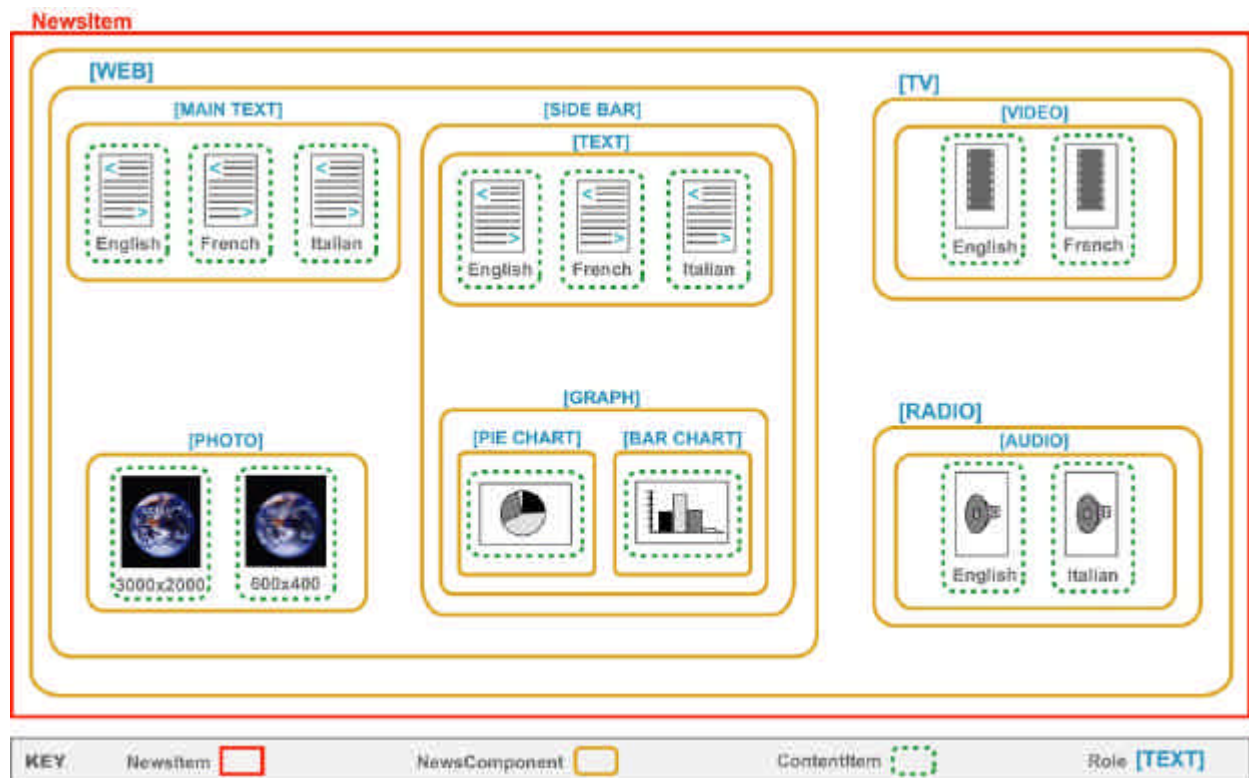
NewsComponentの抽象的な役割を示すものであり、物理的属性を示すことには使わない。

※複数素材の扱いについては、別途「7.3 NewsMLNewsML 複合ガイドライン」を参照。

5.7.1 NewsComponents の動きの図解

次の図は同じ記事をウェブ、テレビ、ラジオ用として作った3つのNewsComponentからなる1つのNewsItemを表している。テレビ、ラジオ版は、それぞれ1つのNewsComponentを含んでいる（それぞれビデオVIDEO、音声AUDIO）。ウェブ版はいくつものNewsComponentから構成されている（主テキストMAIN TEXT、写真PHOTO、サイドバーSIDE BAR）。サイドバーは2つのNewsComponentを持っている（テキスト、グラフGRAPH）。最後に、グラフは、同じ情報を異なる方法で見せた2つのNewsComponentを持っている（円グラフPIE CHART、棒グラフBAR CHART）。

ビデオ、音声、テキスト、主テキストのNewsComponentは、異なる言語の記事を入れたContentItemを含んでいる。写真のNewsComponentは、異なる解像度のContentItemを含んでいる。円グラフ、棒グラフのNewsComponentは、それぞれ1つだけのContentItemを含んでいる。



ここでは、上にイラストで示した例がどのようにNewsML文書で構造化されているかを見せている。

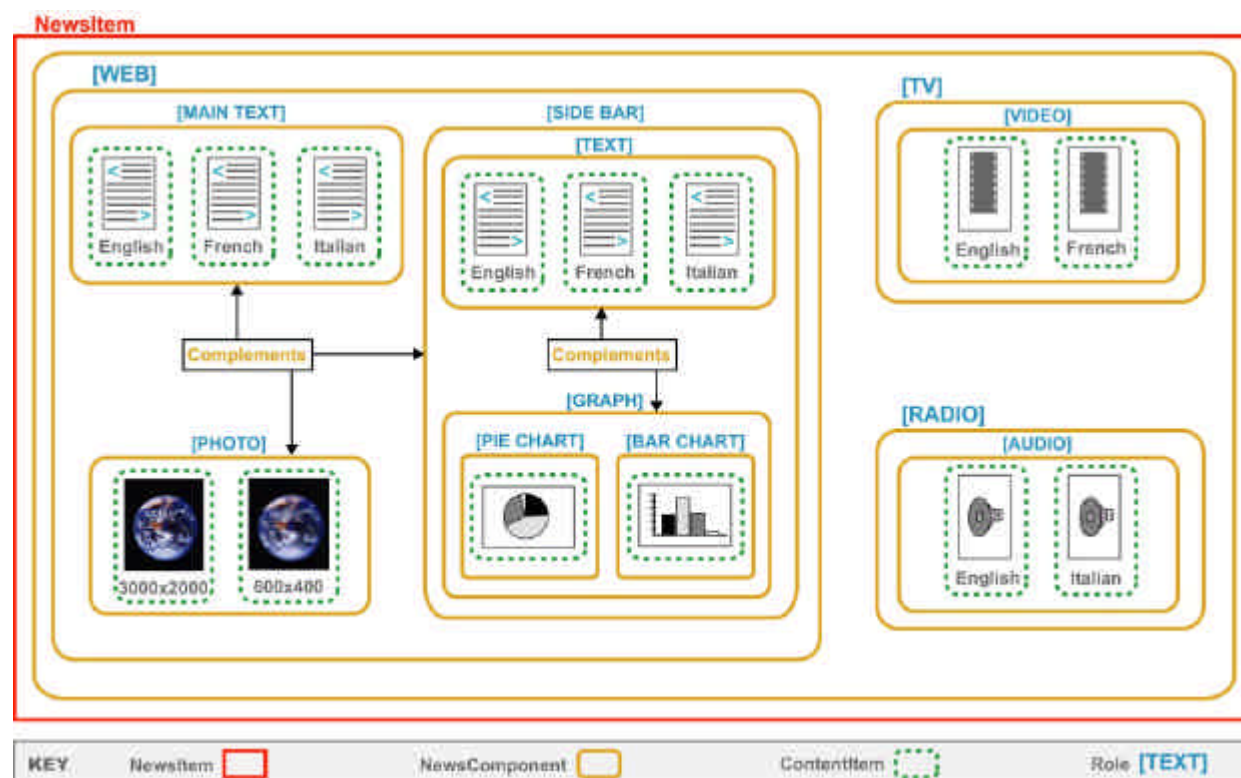
```
<NewsItem>
  <Catalog>
    <Resource>
      <Url>http://www.mysite.com/MyRolesVocabulary.xml</Url>
      <DefaultVocabularyFor Context="Role"/>
    </Catalog>
    ...
    <NewsComponent EquivalentsList="yes">
      <BasisForChoice>./Role/@FormalName</BasisForChoice>
      <NewsComponent EquivalentsList="no">
        <Role FormalName="WEB"/>
        <NewsComponent EquivalentsList="yes">
          <Role FormalName="MAIN TEXT"/>
          <BasisForChoice>./Role/@FormalName</BasisForChoice>
          <ContentItem>...</ContentItem>
```

```

    <ContentItem>...</ContentItem>
    <ContentItem>...</ContentItem>
  </NewsComponent>
  <NewsComponent EquivalentsList="yes">
    <Role FormalName="PHOTO"/>
    <ContentItem>...</ContentItem>
    <ContentItem>...</ContentItem>
    <ContentItem>...</ContentItem>
  </NewsComponent>
  <NewsComponent EquivalentsList="no">
    <Role FormalName="SIDE BAR"/>
    <NewsComponent EquivalentsList="yes" Essential="yes">
      <Role FormalName="TEXT"/>
      <ContentItem>...</ContentItem>
      <ContentItem>...</ContentItem>
      <ContentItem>...</ContentItem>
    </NewsComponent>
    <NewsComponent EquivalentsList="yes" Essential="yes">
      <Role FormalName="GRAPH"/>
<BasisForChoice>./Role/@FormalName</BasisForChoice>
    <NewsComponent>
      <Role FormalName="PIE CHART"/>
      <ContentItem>...</ContentItem>
    </NewsComponent>
    <NewsComponent>
      <Role FormalName="BAR CHART"/>
      <ContentItem>...</ContentItem>
    </NewsComponent>
  </NewsComponent>
</NewsComponent>
<NewsComponent>
  <Role FormalName="TV"/>
  <NewsComponent EquivalentsList="yes">
    <Role FormalName="VIDEO"/>
    <ContentItem>...</ContentItem>
    <ContentItem>...</ContentItem>
  </NewsComponent>
</NewsComponent>
<NewsComponent>
  <Role FormalName="RADIO"/>
  <NewsComponent EquivalentsList="yes">
    <Role FormalName="AUDIO"/>
    <ContentItem>...</ContentItem>
    <ContentItem>...</ContentItem>
  </NewsComponent>
</NewsComponent>
</NewsComponent>
</NewsItem>

```

EquivalentsListであるNewsComponentとEquivalentsList ではないNewsComponentの区別が、次の2つの絵で明らかにされている。



5.7.3 BasisForChoice

[BasisForChoice](#)の構成要素は、[NewsComponent](#)または[ContentItem](#)にある情報を識別する[XPath](#)パターンあるいは要素タイプ名であり、同等の[NewsComponent](#)または[ContentItem](#)を選択する際の基礎として用いられる。[XPath](#)パターンが「.」（ピリオド）で始まっている場合には、[XPath](#)の「ルート（root）」を表し、[NewsComponent](#)あるいは[ContentItem](#)自体に相当する。[XPath](#)パターンを等価の組み合わせの中で順番に個々の[NewsComponent](#)や[ContentItem](#)に適用することにより、システムは項目間の選択を行う基礎となるデータを抽出することができる。[XPath](#)パターンに適合するものが、「root」で始まるサブツリーに多数存在する場合には、文書の順番で最初に適合したもののみが重要である。オプションの[Rank](#)属性により、配信社が異なった[BasisForChoice](#)を付加した場合には、数値により重要度に優先順位を付けることができる。この場合、数値が小さいほど重要度が高くなる。

```
<!ELEMENT BasisForChoice (#PCDATA)>
<!ATTLIST BasisForChoice %localid;
    Rank CDATA #IMPLIED >
```

次の例では、内部[NewsComponent](#)（この場合は円グラフまたは棒グラフ）の役割（[Role](#)）を示している。それらは、[NewsComponent](#)間の選択の基礎として示されている。[BasisForChoice](#)内の「./」（ピリオド＋スラッシュ）は、そのパス（path）のルート（root）の下位要素を示す[XPath](#)シンタックスであり、選択肢となる個々の[NewsComponent](#)を示す。

```
<NewsComponent EquivalentList="yes" Essential="yes">
  <Role FormalName="GRAPH"/> <BasisForChoice>./Role</BasisForChoice>
  <NewsComponent>
    <Role FormalName="PIE CHART"/> <ContentItem>...</ContentItem>
  </NewsComponent>
  <NewsComponent>
    <Role FormalName="BAR CHART"/>
    <ContentItem>...</ContentItem>
  </NewsComponent>
</NewsComponent>
```

次の例では、PixelWidthという値の[FormalName](#)属性を持つ[Property](#)要素の[Value](#)属性が、[ContentItem](#)間の選択の基礎となることを示すために、さらに複雑な[XPath](#)表現を用いる。

```
<Catalog>
  <Resource Duid="resource1">
    <Urn>urn:newsml:mydomain.com:20010101:Characteristics:3</Urn>
    <Url>www.mydomain.com/vocabs/characteristics.xml</Url>
    <DefaultVocabularyFor Context="Property"/>
  </Resource>
</Catalog>
...
<NewsComponent EquivalentList="yes">
  <BasisForChoice>./Property[@FormalName="PixelWidth"]/@Value</BasisForChoice>
  <ContentItem Href="pictures/4769w336.jpg">
    <MimeType FormalName="image/jpeg"/>
    <Characteristics>
      <SizeInBytes>22999</SizeInBytes>
      <Property FormalName="PixelWidth" Value="336"/>
      <Property FormalName="PixelHeight" Value="224"/>
    </Characteristics>
  </ContentItem>
  <ContentItem Href="pictures/4769w170.jpg">
    <MimeType FormalName="image/jpeg"/>
    <Characteristics>
```



```
<SizeInBytes>8449</SizeInBytes>
<Property FormalName="PixelWidth" Value="170"/>
<Property FormalName="PixelHeight" Value="224"/>
</Characteristics>
</ContentItem>
</NewsComponent>
```

Example 2:

以下の[NewsComponent](#)について見てみよう：

```
<NewsComponent EquivalentList="yes">
  <BasisForChoice>@xml:lang</BasisForChoice>
  <NewsComponent xml:lang="en-US">
    ...
  </NewsComponent>
  <NewsComponent xml:lang="fr-FR">
    ...
  </NewsComponent>
</NewsComponent>
```

上の例で、外側の[NewsComponent](#)は、互いに等価である2つの子[NewsComponent](#)を持ち、[BasisForChoice](#)は[xml:lang](#)属性である。

この[BasisForChoice](#)は、[xml:lang](#)属性を探す際に、それぞれの子[NewsComponent](#)の“子”の軸（axis）を利用しなければならないことを示している。このことより、[xml:lang](#)がそれぞれの子[NewsComponent](#)直下の子として存在するため、この例は正しい例だといえる。

上記の[BasisForChoice](#)の構造は次のものと等しいことに注意せよ：

```
<BasisForChoice>./@xml:lang</BasisForChoice>
```

両方のケースにおいて子[NewsComponent](#)の子軸が探されることを示している。

Example 3:

```
<NewsComponent EquivalentList="yes">
<BasisForChoice>@xml:lang</BasisForChoice>

  <NewsComponent>
    <NewsComponent xml:lang="en-US">
      ...
    </NewsComponent>
    </NewsComponent>
    <NewsComponent>
    <NewsComponent xml:lang="fr-FR">
      ...
    </NewsComponent>
  </NewsComponent>
</NewsComponent>
```

上記では、[BasisForChoice](#)は[BasisForChoice](#)要素を含む[NewsComponent](#)のそれぞれの子[NewsComponent](#)の“子”

の軸 (axis) を使うべきであることを示している。しかし、この例では、それぞれの子 [NewsComponent](#) は、その直下の子として、[xml:lang](#) 属性を持っていない。これでは [BasisForChoice](#) 要素は正しく構成されない。この例では [xml:lang](#) 属性は子 [NewsComponent](#) の子孫として現れている。したがって [BasisForChoice](#) は次のように構成されなければならない。

```
<BasisForChoice>./@xml:lang</BasisForChoice>
```

"/" は子 [NewsComponent](#) の子孫軸を探さなければいけないことを示す。

[NSK 解説]

[BasisForChoice](#) 要素は、同等の [NewsComponent](#) 要素または [ContentItem](#) 要素を識別する際に、識別すべき情報の [XPath](#) パターンあるいは要素タイプ名を示す。

[NewsComponent](#) 要素の [EquivalentsList](#) 属性が yes に設定されている場合、その子要素の [NewsComponent](#) 要素または [ContentItem](#) 要素の意味は同等であることを示し、これを識別するための [XPath](#) パターンあるいは要素タイプ名を [BasisForChoice](#) 要素の内容として記述する。

[BasisForChoice](#) 要素は属性値として [localid](#) エンティティ、すなわち [Duid](#) 属性と [Euid](#) 属性と [Rank](#) 属性を持つことができる。

[Rank](#) 属性は [BasisForChoice](#) が複数あった場合の優先順位を指定するために使用する。

例) 複数の [BasisForChoice](#) を使用して対象の絞り込みを行う場合

以下の例では、[BasisForChoice](#) を 2 つ記述することにより、[Rank](#) 属性の値に従って、対象の絞り込みが可能になっている。[Rank="1"](#) では、[Role](#) の値を対象としており、Main 画像と Thumbnail 画像を区別できる。Main 画像が必要な場合は、ここで処理を終了すれば良い。Thumbnail 画像が必要な場合は、対象が二つあるので、さらに、[Rank="2"](#) の [Format](#) の値を利用して区別する。

```
<NewsItem>
  <NewsComponent EquivalentsList="yes">
    <BasisForChoice Rank="1">./Role/@FormalName</BasisForChoice>
    <BasisForChoice Rank="2">./Format/@FormalName</BasisForChoice>
    <!-- サムネイル GIF 画像 -->
    <NewsComponent>
      <Role FormalName="Thumbnail"/>
      <ContentItem Href="/Thumbnail.gif">
        <Format FormalName="GIF89a non-interlaced"/>
      </ContentItem>
    </NewsComponent>
    <!-- サムネイル JPEG 画像 -->
    <NewsComponent>
      <Role FormalName="Thumbnail"/>
      <ContentItem Href="/Thumbnail.jpg">
        <Format FormalName="JPEG Baseline"/>
      </ContentItem>
    </NewsComponent>
    <!-- メイン画像 -->
    <NewsComponent>
      <Role FormalName="Main"/>
      <ContentItem Href="/MainPhoto.jpg">
        <Format FormalName="JPEG Baseline"/>
      </ContentItem>
    </NewsComponent>
  </NewsComponent>
</NewsItem>
```

5.7.4 NewsComponent の他の下位要素

[NewsComponent](#)は、オプションの[NewsLines](#)、[AdministrativeMetadata](#)、[RightsMetadata](#)、[DescriptiveMetadata](#)の各要素を含んでもよい。これらの要素の機能は、本文の[5.9Metadata](#)に述べられている。また、[NewsComponent](#)は、NewsML 仕様書に定義されていない、ユーザーが定義するいくつかのメタデータを含む[Metadata](#)要素をいくつ含んでもよい。

5.8 ContentItem の構造

[ContentItem](#)は、人への公開を意図した表現内容（テキスト、イメージ、映像、音声など）を伝える、あるいは識別するための[news object](#)である。NewsML はメディアに依存しない記述言語であるということが特徴で、あらゆる人の感覚（視覚、聴覚、触覚あるいはこれらの組み合わせを含む）に対して、あらゆるメディアを通して表現が可能である。[ContentItem](#)に格納するテキストの推奨フォーマットはIPTC-NAA NITF である。

[ContentItem](#)は、ある素材データを[DataContent](#)要素の中を含むか、あるいは[ContentItem](#)要素の[Href](#)属性を用いて素材データを指し示して伝えなければならない。素材データを指し示すポインタが使用される場合でも、NewsML 文書は素材データが直接含まれる場合と全く同じように解釈される。ポインタを使用する主な理由は、物理的な転送や蓄積に必要なデータ量を削減することや、整形式（well-formed）XML 文書の中で直接含むことができないデータオブジェクトのフォーマットを操作することにある。

[DataContent](#)要素は、エンコード方法を示す1つ以上の[Encoding](#)要素に包含されることがある。[DataContent](#)要素の素材データが含まれる場合には、整形式XML化を中止したり、NewsML DTD に従うことをやめたりして、NewsML 文書が破綻しないように保証する注意が必要である。この問題が起きないように保証する技術的要件は以下の通りである。

- [XML](#)マークアップとして判定される文字列をデータに含まないことを保証すること。
- [XML](#)マークアップとして判定されるいくつかの文字列があった場合でも、XML1.0 仕様書にあるように、特定の文字列]]>をどこにも含まず、CDATA セクションでデータを包含すること。
- データが[XML](#)コンテンツで構成され、NewsML の要素名と同じ名前の要素を持たない場合には、文書の内部DTDサブセットに使用される要素タイプを宣言すること。

[ContentItem](#)の下位要素であるオプションの[MediaType](#)、[MimeType](#)、[Format](#)、[Notation](#)は、それぞれメディア・タイプ、[MIME](#)タイプ、フォーマット、記法（[notation](#)）を表している。その意味とこれらの要素の[FormalName](#)属性に許可された値は、[Vocabulary](#)属性と[Scheme](#)属性によって識別される controlled vocabularies により決定される。

[ContentItem](#)はまた、その物理的特徴についての情報を提供する[Characteristics](#)要素を含むことができる。[Characteristics](#)要素の目的は、解釈前後のデータを扱うため必要なシステム要件の決定を補助することである。つまり、ファイルサイズで言えば、（ラスター・イメージに対しては）縦横のピクセル数、（ビデオ・クリップに対しては）フレーム数、（音声ファイルに対しては）持続時間、（すべての対象に対しては）バイト数をカバーすることができる。NewsML1.0 版では、特定の要素タイプとして[SizeInBytes](#)のみが特性として規定されている。他のすべての特性としては、一般的な[Property](#)要素が使用されている。この一般的な要素の使用に関する説明については、[5.6.11 Property](#)の項を参照のこと。

```
<!ENTITY % data " (Encoding | DataContent )?">

<!ELEMENT Encoding %data; >
<!ATTLIST Encoding %localid;
           Notation CDATA #REQUIRED >

<!ELEMENT DataContent ANY>
<!ATTLIST DataContent %localid; >

<!ELEMENT ContentItem (Comment* , Catalog? , MediaType? , Format? , MimeType? , Notation? ,
Characteristics? , %data; )>
<!ATTLIST ContentItem %localid;
           Href CDATA #IMPLIED >

<!ELEMENT MediaType EMPTY>
```

```

<!ATTLIST MediaType %localid;
                %formalname; >

<!ELEMENT Format EMPTY>
<!ATTLIST Format %localid;
                %formalname; >

<!ELEMENT MimeType EMPTY>
<!ATTLIST MimeType %localid;
                %formalname; >

<!ELEMENT Notation EMPTY>
<!ATTLIST Notation %localid;
                %formalname; >

<!ELEMENT Characteristics (SizeInBytes?, Property*) >
<!ATTLIST Characteristics %localid; >

<!ELEMENT SizeInBytes (#PCDATA) >
<!ATTLIST SizeInBytes %localid; >

```

この例は、コンテンツを解凍するために、unbinhex (binhexのデコード) を行ってから、unzipを行う必要のあるインライン・データを載せている。

```

<ContentItem>
  <Encoding Notation="binhex">
    <Encoding Notation="zip">
      <DataContent>A873B6FE ...</DataContent>
    </Encoding>
  </Encoding>
</ContentItem>

```

この例は、NewsML 1.0の承認に関するIPTC文書の第2版の中でDuidがitem2であるContentItemを参照により再利用するContentItemを示している。このContentItemは、メディアタイプがText、TTNITF フォーマット、MIME-type text/vnd.IPTC.NITFで、ノーテーションがNITFである。2736バイト長で、myproperties.xmlで定義されたWordCount プロパティが450の値を持つ。ノーテーションを解するXMLプロセッサでオブジェクトが扱えるように、NewsML文書の内部サブセット内で、NITFノーテーションが公式に宣言されている。

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE NewsML PUBLIC "urn:newsml:iptc.org:20001006:NewsMLv1.0.dtd:1"
"http://www.iptc.org/NewsML/NewsMLv1.0.dtd"
[
  <!NOTATION NITF PUBLIC "-//IPTC-NAA//DTD NITF-XML 1.0//EN">
]
<NewsML>
  <Catalog>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:IptcMediaTypes.xml</Urn>
      <DefaultVocabularyFor Scheme="IptcMediaTypes" Context="MediaType"/>
    </Resource>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:IptcFormats.xml</Urn>
      <DefaultVocabularyFor Scheme="IptcFormats" Context="Format"/>
    </Resource>
    <Resource>
      <Urn>urn:newsml:iptc.org:20001006:IptcMimeTypes.xml</Urn>
      <DefaultVocabularyFor Scheme="IptcMimeTypes" Context="MimeType"/>
    </Resource>
  </Catalog>

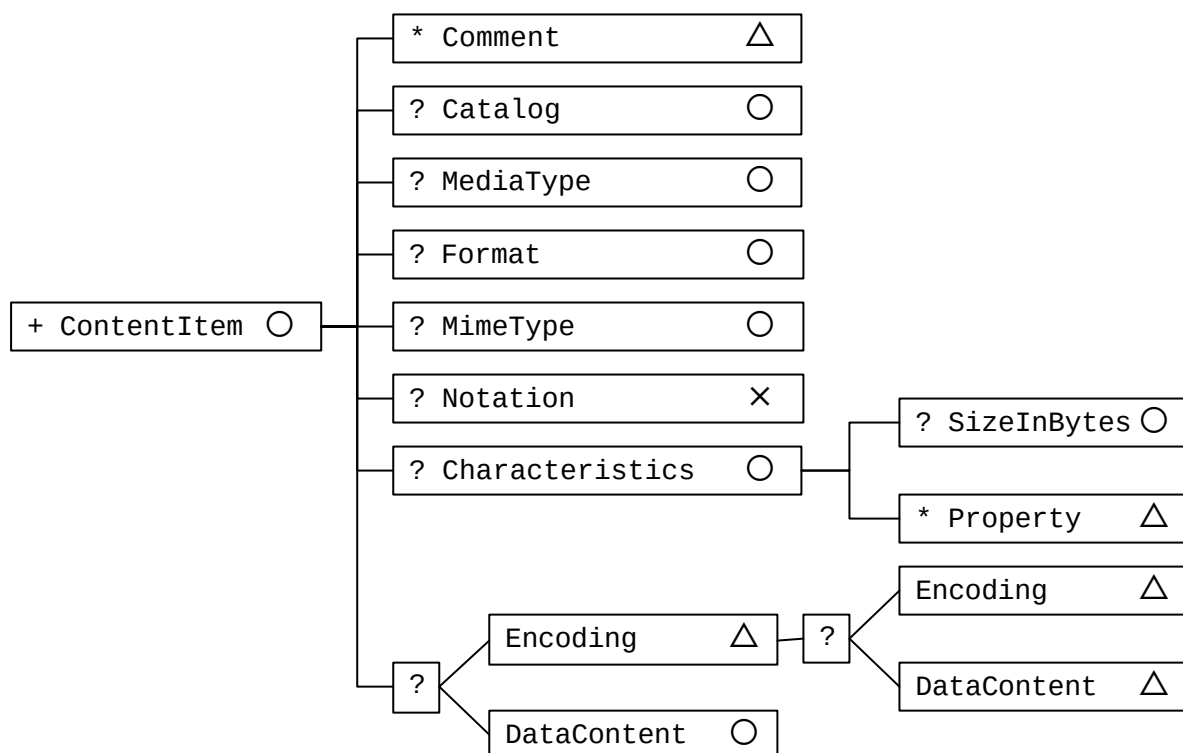
```

```

</Resource>
<Resource>
  <Urn>urn:newsmml:iptc.org:20001006:IptcNotations.xml</Urn>
  <DefaultVocabularyFor Scheme="IptcNotations" Context="Notation"/>
</Resource>
<Resource>
  <Urn>urn:newsmml:mydomain.org:20010101:myproperties.xml</Urn>
  <DefaultVocabularyFor Scheme="Properties" Context="Property"/>
</Resource>
</Catalog>
...
<ContentItem Href="urn:newsmml:iptc.org:20001006:NewsML%201.0%20approved:2#item2">
  <MediaType FormalName="Text"/>
  <Format FormalName="TTNITF"/>
  <MimeType FormalName="text/vnd.IPTC.NITF"/>
  <Notation FormalName="NITF"/>
  <Characteristics>
    <SizeInBytes>2736</SizeInBytes>
    <Property FormalName="WordCount" Value="450"/>
  </Characteristics>
</ContentItem>
...
</NewsML>

```

[ツリー図]



[NSK 解説]

[ContentItem](#)は、実際のコンテンツデータと、そのデータ形式のメタデータを記述する。

コンテンツデータがテキストの場合は、そのまま [DataContent](#)に入れることができる。あるいは NewsML 文書とは別のファイルとして外部におき、[ContentItem](#)要素の[Href](#)属性により参照することもできる。[Href](#)属性で参照した場合には[DataContent](#)要素は含まない。画像、音声、動画等のバイナリデータの場合は[ContentItem](#)要素の[Href](#)属性により外部に置くか、[Encoding](#)要素によって[DataContent](#)に埋め込むことができる。

[Characteristics](#)要素ではコンテンツデータのファイルサイズを表す [SizeInBytes](#)要素と拡張用の [Property](#)要素が規定されており詳しくは、新聞・通信社用 IPTC NewsML 実装ガイドラインに説明がある。

[Notation](#)エンティティは、表現しようとしているニュースオブジェクトがどんな方法で記述、表現されているかを示すために使用される。この要素のオプションには[FormalName](#)属性、[Vocabulary](#)属性、[Scheme](#)属性があり、[FormalName](#)属性に許可された値は、[Vocabulary](#)属性と[Scheme](#)属性によって識別される controlled vocabularies により決定される。

[NskNewsML:1 記述形式]

ContentItem 要素○
ContentItem /Href 属性 ...○
Catalog 要素○

MediaType 要素○
 Format 要素○
 MimeType 要素○
 Notation 要素×
 Characteristics 要素○
 SizeInBytes 要素○
 Property 要素△
 Encoding 要素△
 DataContent 要素○

[IPTC](#)ではテキストのフォーマットとして IPTC-NAA NITF を推奨している。

バイナリデータは外部ファイルとして [ContentItem](#) 要素の [Href](#) 属性 によって参照する形式を推奨する。

[MediaType](#)、[MimeType](#)、Format 要素は組み合わせてデータ型式を記述する。[MimeType](#)は極力使用する。

[MediaType](#) : テキスト、画像、音声、動画など分類を行う

Text	テキスト
Graphic	画像
Photo	写真
Audio	音声
Video	動画
Animation	アニメーション

[MimeType](#) : マイムタイプ

image/jpeg image/tiff text/plain text/xml video/mpeg application/postscript 等。

[Format](#) : [MimeType](#)だけでは表現できない場合に記述することができる。

MimeType="text/xml"のときは、Format は必須。このときの Format のボキャブラリは NSK 拡張ボキャブラリを推奨する。

[Notation](#) : SGML の名残であり、通常は使用しない。

[SizeInBytes](#)は外部ファイルのサイズを示すことができる。

10 進数で記述する。単位は byte だが単位は記述しない。

例) <SizeInBytes>12736</SizeInBytes>

例) プレーンテキストの場合

```

<ContentItem>
  <MediaType FormalName="Text" />
  <MimeType FormalName="text/plain" />
  <DataContent>
    .
    ...プレーンテキスト...
    .
  </DataContent>
</ContentItem>

```

例) XMLコンテンツの場合

```
<ContentItem>
  <MediaType FormalName="Text" />
  <Format FormalName="BML" />
  <MimeType FormalName="text/xml" />
  <DataContent>
    .
    ...XMLコンテンツ...
    .
  </DataContent>
</ContentItem>
```

例) 外部参照の場合

ここではJPEG ファイルの画像フォーマットの場合を示した。

```
<ContentItem Href="20010423photo001.jpg" >
  <MediaType FormalName="Photo" />
  <MimeType FormalName="image/jpeg" />
  <Characteristics>
    <SizeInBytes>12736</SizeInBytes>
    <!-- Width: -->
    <Property Scheme="CharacteristicsProperty" FormalName="Width" Value="3008"/>
    <!-- Height: -->
    <Property Scheme="CharacteristicsProperty" FormalName="Height" Value="2000"/>
    <!-- ColorSpace: -->
    <Property Scheme="CharacteristicsProperty" FormalName="ColorSpace" Value="RGB"/>
    <!-- Resolution: -->
    <Property Scheme="CharacteristicsProperty" FormalName="Resolution" Value="300"/>
  </Characteristics>
</ContentItem>
```


5.9 メタデータ

NewsML は、[NewsComponent](#)において、以下のカテゴリーのメタデータの使用を認めている。

[AdministrativeMetadata](#) (管理メタデータ)

[RightsMetadata](#) (権利メタデータ)

[DescriptiveMetadata](#). (記述メタデータ)

5.9.1 Administrative Metadata (管理メタデータ)

[AdministrativeMetadata](#)要素は、[NewsComponent](#)の起源、出所に関する情報を含んでいる。この情報は、[AdministrativeMetadata](#)要素の直接の親要素である[NewsComponent](#)に、あるいは[NewsComponent](#)の直接の親要素である [NewsItem](#) に適用される。

オプションの[FileName](#)要素は、[NewsItem](#) の仮想または実際の蓄積ファイル名に一致している。

オプションの[SystemIdentifier](#)要素は、その項目が存在する場所の[URL](#)のようなシステムアドレスを指定する。これは、ある意味でXML1.0仕様書に定義された[NewsItem](#) のシステム識別子を規定している。

オプションの[Provider](#)要素と [Creator](#) 要素は、その[news object](#)を公開したり作成した個人及び（または）企業、組織を規定する。(オプションの[Comment](#)で、これに関連した追加情報を付加することができる)

オプションで反復可能な[Source](#)要素は、[news object](#)の素材をどこ(個人および/または企業、組織)が配信したのかを規定する。オプションの [NewsItem](#) 属性は、記事配信の際に [NewsItem](#) に存在しなければならない。そのために、配信された[NewsItem](#) のURNを提供する。[Source](#)要素のシーケンスは、[NewsItem](#) が経由した加盟社のシーケンスを表示することにも使用することができることに注目せよ。ここでも[Comment](#)で、関連した追加情報を付加することができる。

オプションで反復可能な[Contributor](#)要素は、生成された[news object](#)の改版や強化を行った個人及び（または）企業、組織を規定する。この[Comment](#)要素で、それらの関与状況を示すこともできる。

[Creator](#)要素および[Contributor](#)要素に対するオプションで反復可能な[Contribution](#)要素は、[news object](#)の作成や修正を行った[Creator](#)や[Contributor](#)の関与の種別を規定する。

オプションで反復可能な[Property](#)要素は、NewsML DTD 内では明確に与えられていない付加的な[AdministrativeMetadata](#)を提供するのに用いることができる。

```

<!ELEMENT AdministrativeMetadata (Catalog? , FileName? , SystemIdentifier? , Provider? ,
Creator* , Source* , Contributor* , Property* )>
<!ATTLIST AdministrativeMetadata %localid; >

<!ELEMENT FileName (#PCDATA )>
<!ATTLIST FileName %localid; >

<!ELEMENT SystemIdentifier (#PCDATA )>
<!ATTLIST SystemIdentifier %localid; >

<!ELEMENT Provider (%party;) >
<!ATTLIST Provider %localid; >

<!ELEMENT Creator (%party;, Contribution*) >
<!ATTLIST Creator %localid; >

<!ELEMENT Source (%party;) >
<!ATTLIST Source %localid;
                NewsItem CDATA #IMPLIED >

<!ELEMENT Contributor (%party;, Contribution*) >
<!ATTLIST Contributor %localid; >

<!ELEMENT Contribution EMPTY>
<!ATTLIST Contribution %localid; %formalname;>

```

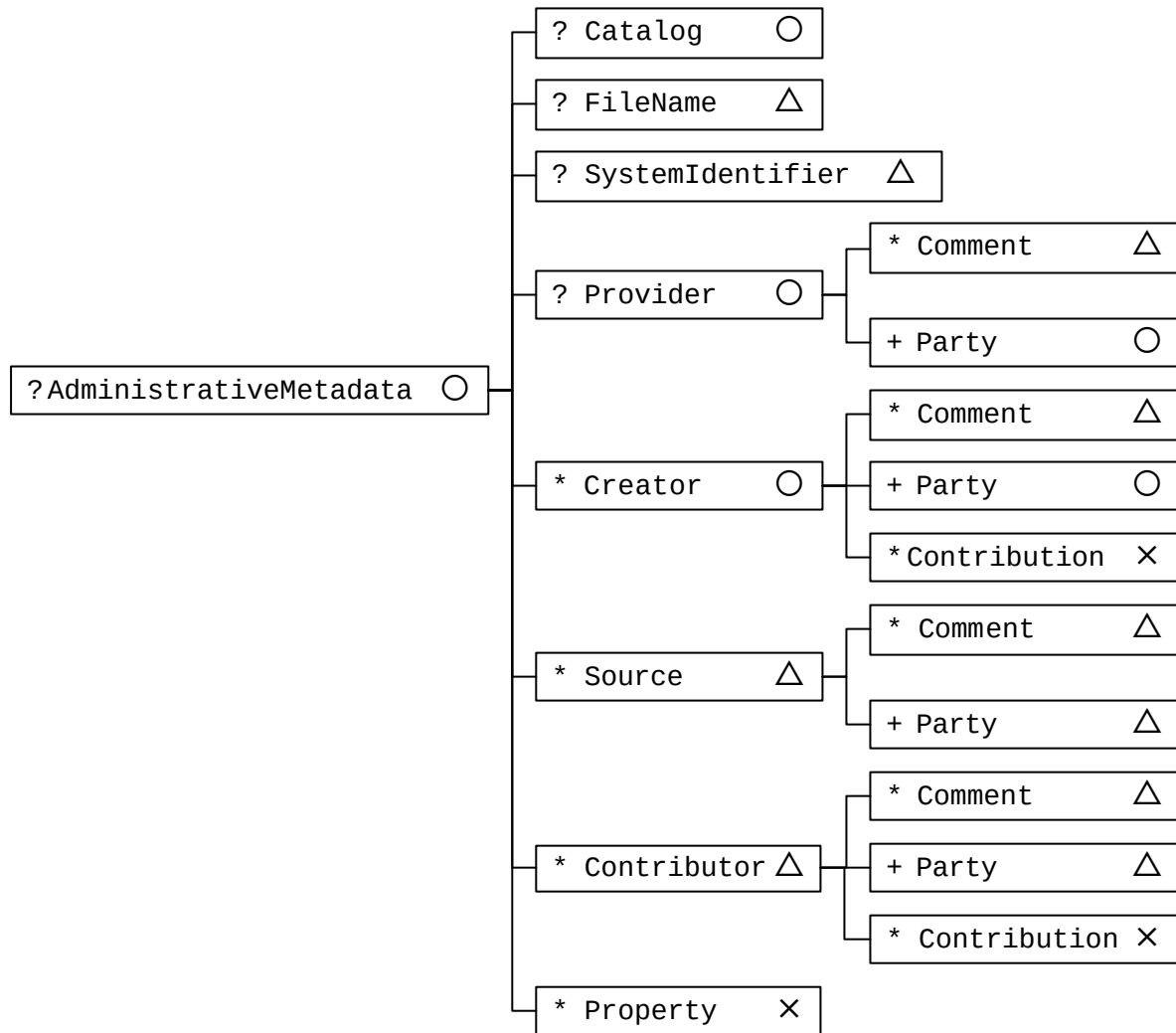
この例では、ファイル名が NewsmlStory.xml で、www.mydomain.com のサブディレクトリ stories にある。配信社は、現在の文書内で [Duid](#) 属性の値が company1 である [Topic](#) 要素によって表される会社である。作成者は、現在の文書内で [Duid](#) 属性の値が person1 である [Topic](#) 要素によって表される。関与した人が2人いて、ひとは編集校閲者で、もう1人は引用されている。彼らはそれぞれ、現在の文書内で [Duid](#) 属性の値が person2 と person3 である [Topic](#) 要素によって表される。

```

<AdministrativeMetadata>
  <FileName>NewsmlStory.xml</FileName>
  <SystemIdentifier>http://www.mydomain.com/stories/NewsmlStory.xml</SystemIdentifier>
  <Provider>
    <Party FormalName="News Headlines International" Topic="#company1"/>
  </Provider>
  <Creator>
    <Party FormalName="Doe, John" Topic="#person1"/>
  </Creator>
  <Contributor>
    <Comment>Editorial review</Comment>
    <Party FormalName="Smith, Jane" Topic="#person2"/>
  </Contributor>
  <Contributor>
    <Comment>Quote</Comment>
    <Party FormalName="Dumas, Pierre" Topic="#person3"/>
  </Contributor>
</AdministrativeMetadata>

```

[ツリー図]



[NSK 解説]

[Provider](#)要素は、[NewsComponent](#)の配信社を示す。

[Creator](#)要素は、[NewsComponent](#)の作成社を示す。

[Provider](#)要素と[Creator](#)要素の[Party](#)要素には個別のボキャブラリを使用してよい。

(例) A 社が作った NewsComponent を A 社が B 社に配信したデータの場合

```

<AdministrativeMetadata>
  <Provider>
    <Party FormalName="Acompany"/>
  </Provider>
  <Creator>
    <Party FormalName="Acompany"/>
  </Creator>
</AdministrativeMetadata>
  
```

(例) 上記をさらに B 社が配信したデータの場合

```
<AdministrativeMetadata>
  <Provider>
    <Party FormalName="Bcompany"/>
  </Provider>
  <Creator>
    <Party FormalName="Acompany"/>
  </Creator>
</AdministrativeMetadata>
```

[FileName](#)要素は、NewsItemすなわち NewsItemTypeで表される様々な種別の情報（記事・写真・スポーツの試合結果・株価・[TopicSet](#)ファイル・Catalog ファイル等々）を指し示すこと、またはそれらが実際に保存されているファイル名を表すことができる。

[FileName](#)要素の設定値は、文字データを表す「#PCDATA」であり、0または1個の記述ができる。

[SystemIdentifier](#)要素は、NewsItemのシステム的な識別子であり、[URL](#)などによって記述される。その意味/意義についてはXML1.0にて定義されており、0または1個の記述ができる。

[Source](#)要素は、[NewsComponent](#)の情報源（ソース）または素材を示すことができる。さらに子要素として[Party](#)要素が1つ以上含まれ、これにより情報源となった個人、企業、組織（配信社など）がわかる。同じく子要素の[Comment](#)要素に説明を追加することもできる。

```
<Source>
  <Party FormalName="Acompany"/>
</Source>

<Source>
  <Comment>A- COMPANYは・・・ </Comment>
  <Party FormalName="Acompany"/>
  <Party FormalName="Bcompany"/>
</Source>
```

素材となったのがNewsMLのコンテンツ（厳密にはNewsItem）なら、NewsItem属性にそのURNを記述することができる。

```
<Source NewsItem="urn:newsml:provider.or.jp:20031002:xx001">
  <Party FormalName="Acompany"/>
</Source>
```

[Contributor](#)要素は、NewsComponentの修正、追加を行なった個人、組織等を示す。

[Contributor](#)要素は複数回使用することが可能で[Comment](#)要素を使用することで、関与の内容についても記述することができる。

[Creator](#)要素下の[Contribution](#)は、この要素を持つ[NewsComponent](#)の作成にあたっての役割を書くもので、通常、複数の人や組織を書き並べる場合に利用する。

[Contributor](#)要素下の[Contribution](#)は、この要素を持つ[NewsComponent](#)の作成あるいは修正時に関与した場合に、その関与に当たっての役割を書くもので、通常、複数の人や組織を書き並べる場合に利用する。

[NewsML 仕様の変更点：NewsML v1.2]

[Creator](#)要素が繰り返し可能となった。

[Creator](#)と[Contributor](#)に繰り返し可能な[Contribution](#)要素が追加された。

[NskNewsML:1 記述形式]

AdministrativeMetadata 要素 …○
Catalog 要素 ……………○
Provider 要素 ……………○
Provider/Party 要素 ……………○
Creator 要素 ……………○
Creator/Party 要素 ……………○
Property 要素 ……………×

[NskNewsML:1.2 記述形式]

AdministrativeMetadata 要素 …○
Catalog 要素 ……………○
Provider 要素 ……………○
Provider/Party 要素 ……………○
Creator 要素 ……………○
Creator/Party 要素 ……………○
Creator/Contribution 要素 ……×Contributor/Contribution 要素 ・×Property 要素 ……………×

Creator 要素が繰り返し可能となったが、1.0 と同様 0 回または 1 回の記述とする。Creator 要素と Contributor 要素の子要素として、繰り返し可能な Contribution 要素が追加されたが、使用停止とする。

5.9.2 Rights Metadata (権利メタデータ)

[RightsMetadata](#)要素は、[NewsComponent](#)に関する権利についての情報を持っており、著作権保持者が他者に与えた関連の使用権を含む。

[Copyright](#)要素は、必須の[CopyrightHolder](#)、[CopyrightDate](#)下位要素、およびオプションで繰り返し可能な[Comment](#)の下位要素を持つ。[assignment](#)属性は、著作権がだれによって指定されたか、どの程度の重要性和信頼性を持っているか、ISO8601 基本フォーマットに基づく日付と時間などを表示する。[CopyrightHolder](#)、[CopyrightDate](#)要素には、著作権保持者と著作権日付を自由形式で記述する。

[RightsMetadata](#)は、テキスト中にオプションとして[Origin](#)要素が点在しているものを内容とする下位要素を含んでいる。テキストの内容は人間が理解できるものになっている。[Origin](#)要素はこのテキストの全部あるいは一部を包むものであり、そこに自然言語で述べられていることに形式的に対応したデータの項目へのポインタを与える。[Origin](#)要素の[Href](#)属性は、関連するデータを識別する。そしてその[Href](#)属性はhttp URL か NewsML URN であり、オプションとして[fragment identifier](#)を続けて持つ。他の方法としては、現在の文書の中で「#」(シャープ)記号に要素の[Duid](#)が続くシンプルな[fragment identifier](#)であってもよい。

しかしながら、[Origin](#)要素はシステム処理用のデータを与えており、別の場所で、同じ情報が人間に分かるテキストとして伝えられる。[RightsMetadata](#)の[UsageRights](#)下位要素は、[NewsComponent](#)に属する使用権についての情報を提供する。[UsageRights](#)要素は以下の6つの下位要素から構成されている。使用権を適用する使用タイプを自由言語で示す[UsageType](#)、特定の使用権に関係する地域や場所を示す[Geography](#)、使用権所有者を示す[RightsHolder](#)、[NewsComponent](#)の内容の使用に関する制限を示す[Limitations](#)、最後に、示された権利の期間を示す[StartDate](#)と[EndDate](#)である。

オプションで繰り返し可能な[Property](#)要素は、NewsML DTD の中では明示されない任意の付加的な権利メタデータを記述するために使われる。

```
<!ELEMENT RightsMetadata ( Catalog? , Copyright* , UsageRights* , Property* )>
<!ATTLIST RightsMetadata %localid;
                    %assignment; >

<!ELEMENT Copyright ( Comment* , CopyrightHolder , CopyrightDate )>
<!ATTLIST Copyright %localid;
                    %assignment; >

<!ELEMENT CopyrightHolder (#PCDATA | Origin)*>
<!ATTLIST CopyrightHolder %localid;
                    xml:lang CDATA #IMPLIED >

<!ELEMENT CopyrightDate (#PCDATA | Origin)*>
<!ATTLIST CopyrightDate %localid;
                    xml:lang CDATA #IMPLIED >

<!ELEMENT UsageRights ( UsageType? , Geography? , RightsHolder? , Limitations? , StartDate? ,
EndDate? )>
<!ATTLIST UsageRights %localid;
                    %assignment; >

<!ELEMENT UsageType (#PCDATA | Origin)*>
<!ATTLIST UsageType %localid;
                    xml:lang CDATA #IMPLIED
                    %assignment; >

<!ELEMENT Geography (#PCDATA | Origin)*>
<!ATTLIST Geography %localid;
```

```

        xml:lang CDATA #IMPLIED
        %assignment; >

<!ELEMENT RightsHolder (#PCDATA | Origin)*>
<!ATTLIST RightsHolder %localid;
        xml:lang CDATA #IMPLIED
        %assignment; >

<!ELEMENT Limitations (#PCDATA | Origin)*>
<!ATTLIST Limitations %localid;
        xml:lang CDATA #IMPLIED
        %assignment; >

<!ELEMENT StartDate (#PCDATA | Origin)*>
<!ATTLIST StartDate %localid;
        xml:lang CDATA #IMPLIED
        %assignment; >

<!ELEMENT EndDate (#PCDATA | Origin)*>
<!ATTLIST EndDate %localid;
        xml:lang CDATA #IMPLIED
        %assignment; >

<!ELEMENT Origin (#PCDATA | Origin)*>
<!ATTLIST Origin %localid;
        %assignment;
        Href CDATA #IMPLIED >

```

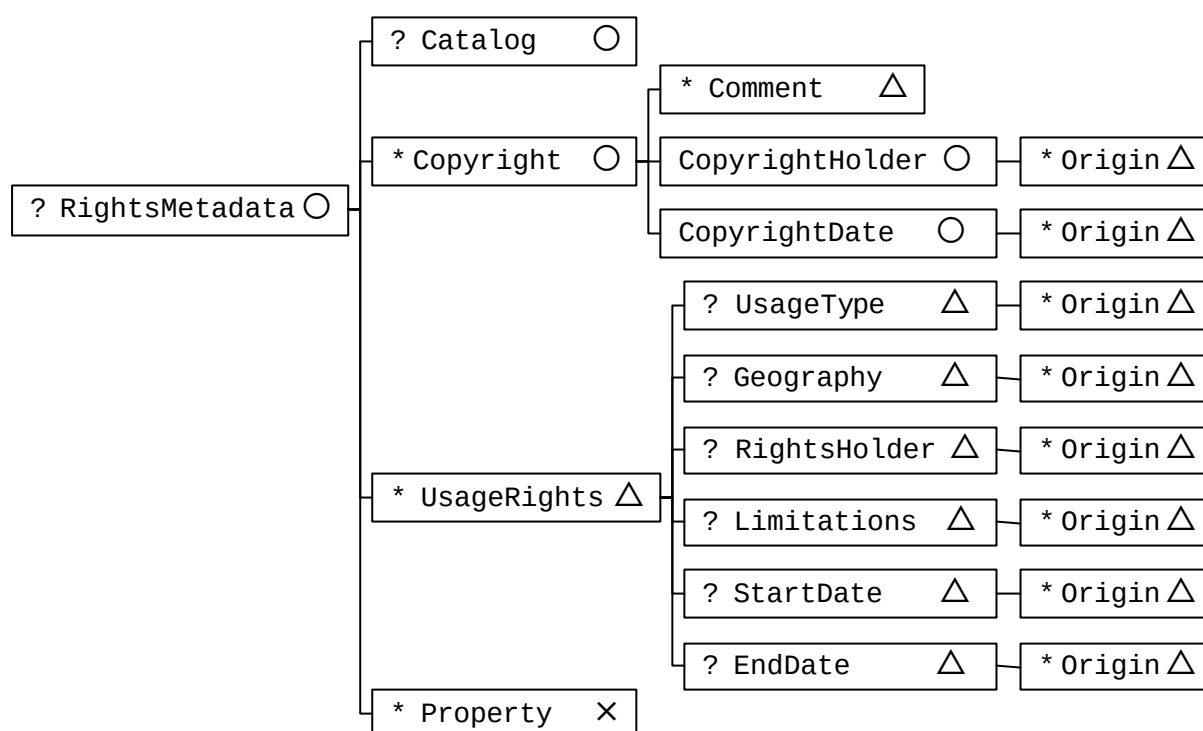
次の例では、[Origin](#)要素が、現在の文書中のTopicsを参照することによって、会社、組織、地域を識別する。国（英国）がIPTC Countries TopicSetを参照することによって識別されている。この[TopicSet](#)はISO 2文字・3文字国コード・ネーミング・スキームに一致する[controlled vocabulary](#)として働く。

```

<RightsMetadata>
  <Copyright>
    <CopyrightHolder><Origin Href="#organization1">International Press
Telecommunications Council</Origin></CopyrightHolder>
    <CopyrightDate>2000</CopyrightDate>
  </Copyright>
  <UsageRights>
    <UsageType>Television</UsageType>
    <Geography><Origin Href="urn:newsml:iptc.org:20001006:Countries#isoc826">United
Kingdom</Origin></Geography>
    <RightsHolder><Origin Href="#organization2">BBC</Origin></RightsHolder>
    <StartDate>July 2000</StartDate>
    <EndDate>December 2000</EndDate>
    <Limitations>Acknowledgement of <Origin Href="#organization1">IPTC</Origin>
copyright must be made</Limitations>
  </UsageRights>
  <UsageRights>
    <UsageType>Television</UsageType>
    <Geography><Origin Href="#region1">North America</Origin></Geography>
    <RightsHolder><Origin Href="#company1">CNN</Origin></RightsHolder>
    <StartDate>July 2000</StartDate>
    <EndDate>none</EndDate>
    <Limitations>Acknowledgement of <Origin Href="#organization1">IPTC</Origin>
copyright must be made</Limitations>
  </UsageRights>
</RightsMetadata>

```

[ツリー図]



[NSK 解説]

[Copyright](#)要素は 3 つの下位要素により[NewsComponent](#)の著作権(著作権)に関するメタデータを記述する。[Copyright](#)要素自身は 0 回以上複数回記述できる。

下位要素の[Comment](#)要素は 0 回以上複数回、[CopyrightHolder](#)要素、[CopyrightDate](#)要素は必ず 1 回記述する。それぞれの要素はこの順に出現する必要がある。

[CopyrightHolder](#)要素：著作権（著作権）の保持者（社）

[CopyrightDate](#)要素：著作権（著作権）の発生日付

[Origin](#)要素は、自由形式で設定された文字列を規定するために使う。この要素は、[UsageRights](#)要素と [NewsLines](#) 要素のすべての下位要素および[CopyrightHolder](#)要素、[CopyrightDate](#)要素に使用でき、[Href](#)属性を使って、設定された文字列を識別するための Topicset を指定することができる。

[UsageRights](#)要素は、6 つの下位要素により、[NewsComponent](#) の使用権に関するメタデータを記述する。[UsageRights](#)要素自身は 0 回以上複数回記述できる。[UsageRights](#)要素を著作権、肖像権などの権利と混同しないように注意すること。

下位要素の[UsageType](#)要素、[Geography](#)要素、[RightsHolder](#)要素、[Limitations](#)要素、[StartDate](#)要素、[EndDate](#)要素は、0 もしくは 1 回記述でき、この順に出現する必要がある。

[UsageType](#)要素：テレビ、新聞、ウェブなど使用権が適応されるメディアの種別

[Geography](#)要素：使用権が適用される地域

[RightsHolder](#)要素：使用権の保持者（社）

[Limitations](#)要素：使用時の制限事項

[StartDate](#)要素と[EndDate](#)要素：使用権の期限

例 [xml:lang](#)属性により言語を指定して記述することができる。

```
<Copyright>
  <CopyrightHolder xml:lang="ja-JP">A 通信社</CopyrightHolder>
  <CopyrightDate xml:lang="ja-JP">平成 1 3 年</CopyrightDate>
</Copyright>
```

コラム：著作権と使用権の表現場所

著作権と使用権に関する情報は、NewsML では[NewsLines](#)（[CopyrightLine](#)要素、[RightsLine](#)要素）と[RightsMetadata](#)（[Copyright](#)要素、[UsageRights](#)要素）のそれぞれ記述できる。

[NewsLines](#)は人が見てわかる形式での情報を、[RightsMetadata](#)にはメタ形式での表現で記述ことになる。[NewsLines](#)、[RightsMetadata](#)双方に著作権、使用権を記述する場合には、同一の情報を記述することが望ましい。

しかし、これらの情報がいずれも、任意の記述情報であり、制限されない自由形式での表現が可能のため、実際にデータ交換する際は、あらかじめ当事者同士で記述方法、記述内容に関する合意が使用の前提条件となることに注意したい。

[NskNewsML:1記述形式]

RightsMetadata 要素○
Catalog 要素○
Copyright 要素○
Copyright/CopyrightHolder 要素○
Copyright/CopyrightHolder/@xml:lang○
Copyright/CopyrightDate 要素○
Copyright/CopyrightDate/@xml:lang○
Property 要素×

CopyrightHolder は著作権の保有者を自由形式で記述する。

CopyrightDate は著作権が発生した日付を自由形式で記述する。

5.9.3 Descriptive Metadata (記述メタデータ)

[DescriptiveMetadata](#)要素は[NewsComponent](#)の内容を記述する情報を含んでいる。[Language](#)、[Genre](#)、[SubjectCode](#)、[OfInterestTo](#)、[TopicOccurrence](#)、[Property](#)の各下位要素は、[NewsComponent](#)のジャンル、主題、対象視聴者、どの言語を使うといったこと（これは特定の視聴者や出版物に適切かどうかを判断するのに役立つ）を示し、任意の人々、場所、組織、国についての情報や、ニュースが示した他の実世界の事柄、誰にそのニュースが関連するかという情報を与える。

[Language](#)要素は[ContentItem](#)で使われた言語を示している。[FormalName](#)属性の値は[Language](#)要素の形式名である。この値をコントロールする Topicset が定義されない場合は、RFC3066 を特殊な [controlled vocabulary](#) の 1 つとして扱い、RFC3066 に定義されている値を許すことを意味する。

[Genre](#)要素は[NewsComponent](#)のジャンルを示している。[FormalName](#)属性の値は[Genre](#)の形式名である。[Vocabulary](#)要素と[Scheme](#)要素により示された[controlled vocabulary](#)で決まった値を許すことを意味する。

[SubjectCode](#)要素は、[IPTC](#)のサブジェクトコードを含む。それはIPTC Subject Codes TopicSet で定義されていて、NewsItem の主題を示す。1 つ以上の[Subject](#)、[SubjectMatter](#)、[SubjectDetail](#)要素からなり、オプションとして1 つ以上の[SubjectQualifier](#)要素によって拡大される。

[OfInterestTo](#)要素は NewsItem の対象視聴者を示す。[Relevance](#)下位要素は対象視聴者への NewsItem の適合性を示す。[FormalName](#)属性の値は、[OfInterestTo](#)、[Relevance](#)要素の形式名を表し、そのことは[Vocabulary](#)要素と[Scheme](#)要素により示された[controlled vocabulary](#)で決まった値を許すことを意味する。

[DateLineDate](#)要素は NewsItem の作られた論理的な日付を提供する。その内容はISO8601 ベースのクフォーマットである。

[Location](#)要素はその子[Property](#)要素を使って場所情報を表す。オプションの[HowPresent](#)属性は場所の種類を表す。

[TopicOccurrence](#)要素は[NewsComponent](#)でどんな[Topic](#)が現れているかを表す。オプションである[HowPresent](#)属性は、その[Topic](#)の出現の性質を示している。[Topic](#)属性の値は、#記号に現在の文書中の[Topic](#)の[Duid](#)属性値を続けることによって、構成しなければならない。

オプションで繰り返し可能な[Property](#)要素は、NewsML DTD の中では明示されない任意の付加的な説明メタデータを与えるために使われる。

誰によって、どの程度の信用度で[DescriptiveMetadata](#)が割り当てられたのかを示す[assignment](#) エンティティの使用には特に気をつけるべきである。割り当て情報は、もしツリーのより低いレベルで新しい割り当て課題要素によってくつがえされなければ、サブツリー全体に受け継がれる。適性度の能力と重要度を含んだ割り当て情報は、どんな詳細のレベルにでも与えることができることに注意すべきである。

```
<!ELEMENT DescriptiveMetadata ( Catalog?, Language*, Genre*, SubjectCode*,
OfInterestTo*, DateLineDate?, Location*, TopicOccurrence*, Property*)>
<!ATTLIST DescriptiveMetadata %localid;
                                %assignment; >

<!ELEMENT Language EMPTY>
<!ATTLIST Language %localid;
                    %formalname;
                    %assignment; >
```

```

<!ELEMENT Genre EMPTY>
<!ATTLIST Genre %localid;
                %formalname;
                %assignment; >

<!ELEMENT SubjectCode ((Subject | SubjectMatter | SubjectDetail), SubjectQualifier*)*>
<!ATTLIST SubjectCode %localid;
                    %assignment; >

<!ELEMENT Subject EMPTY>
<!ATTLIST Subject %localid;
                %formalname;
                %assignment; >

<!ELEMENT SubjectMatter EMPTY>
<!ATTLIST SubjectMatter %localid;
                    %formalname;
                    %assignment; >

<!ELEMENT SubjectDetail EMPTY>
<!ATTLIST SubjectDetail %localid;
                    %formalname;
                    %assignment; >

<!ELEMENT SubjectQualifier EMPTY>
<!ATTLIST SubjectQualifier %localid;
                    %formalname;
                    %assignment; >

<!ELEMENT TopicOccurrence EMPTY >
<!ATTLIST TopicOccurrence %localid;
                    %assignment;
                    Topic CDATA #IMPLIED >

<!ELEMENT OfInterestTo (Relevance?)>
<!ATTLIST OfInterestTo %localid;
                    %formalname;
                    %assignment; >

<!ELEMENT Relevance EMPTY >
<!ATTLIST Relevance %localid;
                    %formalname;
                    %assignment; >

```

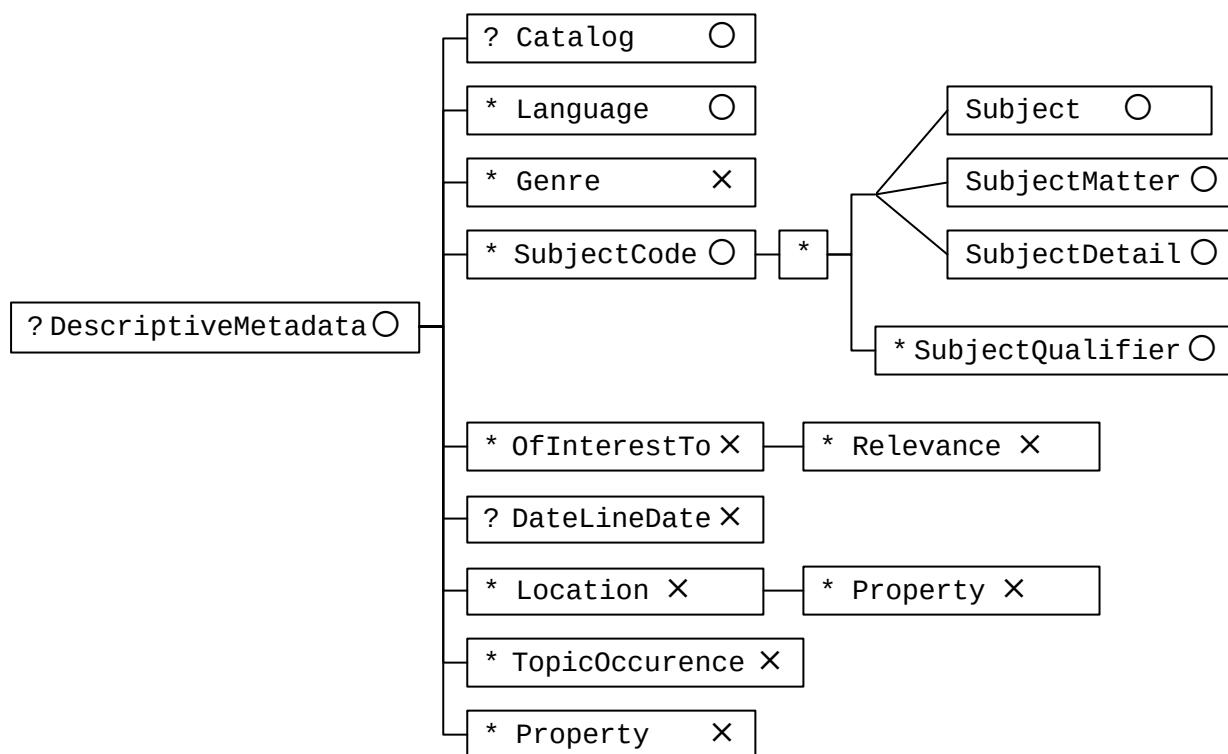
この例では、関連のIPTCボキャブラリが、[TopicType](#)、[Language](#)、[Genre](#)、[Subject](#)、[OfInterestTo](#)要素と、[Confidence](#)、[Importance](#)属性のためのデフォルトとして宣言されている。その時[TopicType](#)は、2人の人物 (Bill ClintonとYasser Arafat)、1つのロケーション(The White House Lawn)を含むと規定されている。それから[DescriptiveMetadata](#)要素が続く。メタデータは信用度 (Confidence) が「高いHigh」、重要度[Importance](#)が「5」(IPTC importance vocabularyでは「普通」と記されている)が割り振られていると宣言している。記述メタデータにより[NewsComponent](#)は英語で記述されていて、ジャンルは「時事 (Current)」、サブジェクトはIPTC Subject「11000000」(IPTC subject codes vocabularyでは政治Politicsと記述)であることがわかる。また、President Clintonの顕著 (Prominent) な出現、White House Lawnのついで (Passing) の出現、Yasser Arafatの関連して (RelatesTo) の出現がわかる。これはWhite House Lawnでサミット会議に出席するYasser Arafatのヘリコプターの到着を待つPresident Clintonの写真からなるTopicOccurrencesの適当なセットである。

```

<Catalog>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcTopicTypes</Urn>
    <DefaultVocabularyFor Scheme="IptcTopicTypes" Context="TopicType"/>
  </Resource>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:Languages</Urn>
    <DefaultVocabularyFor Scheme="IsoLanguageCode" Context="Language"/>
  </Resource>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcGenre</Urn>
    <DefaultVocabularyFor Scheme="IptcGenre" Context="Genre"/>
  </Resource>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcSubjectCodes</Urn>
    <DefaultVocabularyFor Scheme="IptcSubjectCode" Context="Subject"/>
  </Resource>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcOfInterestTo</Urn>
    <DefaultVocabularyFor Scheme="IptcOfInterestTo" Context="OfInterestTo"/>
  </Resource>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcConfidence</Urn>
    <DefaultVocabularyFor Scheme="IptcConfidence" Context="@Confidence"/>
  </Resource>
  <Resource>
    <Urn>urn:newsml:iptc.org:20001006:IptcImportance</Urn>
    <DefaultVocabularyFor Scheme="IptcImportance" Context="@Importance"/>
  </Resource>
</Catalog>
<TopicSet FormalName="Person">
  <Topic Duid="person1">
    <TopicType FormalName="Person"/>
    <Description xml:lang="en-GB">President Clinton</Description>
  </Topic>
  <Topic Duid="person2">
    <TopicType FormalName="Person"/>
    <Description xml:lang="en-GB">Yasser Arafat</Description>
  </Topic>
  <Topic Duid="location1">
    <TopicType FormalName="Location"/>
    <Description xml:lang="en-GB">The White House Lawn</Description>
  </Topic>
</TopicSet>
<DescriptiveMetadata Confidence="High" Importance="5">
  <Language FormalName="en"/>
  <Genre FormalName="Current"/>
  <SubjectCode>
    <Subject FormalName="11000000"/>
  </SubjectCode>
  <TopicOccurrence Topic="#person1" HowPresent="Prominent"/>
  <TopicOccurrence Topic="#person2" HowPresent="RelatesTo"/>
  <TopicOccurrence Topic="#location1" HowPresent="Passing"/>
</DescriptiveMetadata>

```

[ツリー図]



[NSK 解説]

[DescriptiveMetadata](#)要素には、[NewsComponent](#)内にどのような内容が格納されているかを説明する情報を記述する。

ここでは、次のような下位要素を持つ。

[Catalog](#): [DescriptiveMetadata](#)の中で使われるボキャブラリを定義する。使用方法に関しては、「[Catalog](#)」の章を参照。

[SubjectCode](#): [NewsItem](#) の主題を示す。ニュースの分類に使用できる。

複数の[SubjectCode](#)が記述されている場合、また1つの[SubjectCode](#)に複数の「下位要素※」が記述されている場合のいずれの場合にも、[NewsItem](#) が記述されているすべての分類に属することを表す。本要素は、[Subject](#)、[SubjectMatter](#)、[SubjectDetail](#)といった「下位要素※」のいずれかもしくはそれら複数の組み合わせで表現する。

ボキャブラリは[IPTC](#)のボキャブラリを使用する。

[IPTC](#)の[SubjectCode](#)は、3つの「下位要素※」で表現する。それぞれの値は8桁の数値で表現し、先頭2桁が大分類、次の3桁が中分類、最後の3桁が小分類を表現する値となる。【[IPTC](#)の[SubjectCode](#)】

「下位要素※」	意味	値(例)	例の意味
Subject	大分類	15000000	スポーツ
SubjectMatter	中分類	15005000	スポーツ・陸上競技
SubjectDetail	小分類	15005011	スポーツ・陸上競技・10,000m

なお、大分類しか記述しない場合は、[Subject](#)のみで良い。逆に[SubjectDetail](#)は、[Subject](#)、[SubjectMatter](#)

も含んだ値となっているため、[Subject](#)、[SubjectMatter](#)を併記しなくても良い。[SubjectDetail](#)だけの場合でも、[Subject](#)、[SubjectMatter](#)が設定されている場合と同等に解釈する。

また必要に応じて[SubjectQualifier](#)により補足説明することが可能である。

[SubjectQualifier](#)では、[SubjectCode](#)として指定する主題が、スポーツである場合に、男女の別、プロ／アマ、屋内／屋外といった更に詳細な指定を可能とする。上記の例に、以下の[SubjectQualifier](#)を付けると、男子陸上10,000m 競争が表現できる。

「下位要素※」	意味	値(例)	例の意味
SubjectQualifier		補足情報 15000001	男子

(例1) “夏季オリンピック”に関する記述であることを示す場合

```
<SubjectCode>
  <SubjectDetail FormalName="15073001"/>
</SubjectCode>
```

(例2) “夏季オリンピック陸上10,000m 男子決勝”に関する記述であることを示す場合

```
<SubjectCode>
  <SubjectDetail FormalName="15005011"/>
  <SubjectQualifier FormalName="15000001"/>
  <SubjectQualifier FormalName="15000024"/>
  <SubjectDetail FormalName="15073001"/>
</SubjectCode>
```

[Subject](#)、[SubjectMatter](#)、[SubjectDetail](#)、[SubjectQualifier](#)要素の[FormalName](#)属性の値で、[IPTC](#)のボキャブラリに不足するコードの追加に関しては、日本新聞協会を通じて[IPTC](#)に申請できる。

[DescriptiveMetadata](#)要素は[Genre](#)要素と[SubjectCode](#)要素を持つことができ、これら2つの要素はニュースを分類するという目的では共通だが、使用目的と分類方法が異なっていることに注意しなければならない。[SubjectCode](#)要素はNewsItemの主題を、具体的には「夏期オリンピック陸上10,000m男子決勝」のようなカテゴリーを数値化、階層化して示している。一方、[Genre](#)要素は[NewsComponent](#)の事物や表象を、具体的には「時事」「書評」「風景」のように、性質・共通性・本質に着目して分類したジャンルを示す。

[Language](#)要素は、[ContentItem](#)で使用する言語を表現する。コンテンツデータのフォーマットに関わらず、テキスト形式でない動画、音声などの場合も記述することができる。

注意点としては、[Language](#)要素はNewsComponent/ DescriptiveMetadataの子要素であるため、階層が異なり複数の列記が可能な[ContentItem](#)に対して、個別に対応したメタデータとして関連付けができないことである。そのため、[Language](#)要素の値は、[NewsComponent](#)の下位要素に含まれる[ContentItem](#)に対して一括したメタデータとなる。更に[Language](#)要素の出現回数は0回以上になっているため1～複数の[ContentItem](#)に対して1～複数の[Language](#)要素が記述可能になっている。しかし、[Language](#)要素と[ContentItem](#)との関係は表現できないため、この場合もいずれかの[ContentItem](#)において、どれかの（もしくはすべて）の[Language](#)要素で表現する言語が使用されていることを示すに過ぎない。複数の[Language](#)要素と1個の[ContentItem](#)、またその逆も可能である。

[Language](#)要素は、[FormalName](#)属性で値を記述する。

(例) <Language FormalName="ja"/>

[OfInterestTo](#)要素は、NewsItem のターゲットとなる読者や視聴者を示す。その子要素の[Relevance](#)は、

ターゲットとなる読者や視聴者に与えられた NewsItem に対する関係性（適合性）を示す。[OfInterestTo](#) 要素と [Relevance](#) 要素の [FormalName](#) 属性の値は、[Vocabulary](#) と [Scheme](#) 属性で識別される [Controlled Vocabulary](#) によって決定された意味と許可された値で形式名を与える。

DateLineDate 要素は/NewsML/NewsItem/NewsComponent/NewsLines/DateLine と論理的に等しい内容を記述することができる。ここで[DateLine](#)要素は人が読むことができれば記述形式は問わないが、DateLineDate 要素は ISO8601 ベーシックフォーマットで記述する必要がある点に、注意が必要である。

```
<DateLineDate>20030909</DateLineDate>
```

[Location](#) 要素は場所に関する情報を、任意に繰り返すことができる [Property](#) 要素を使って表現する。[HowPresent](#) 属性を付加することによってどのような場所情報を表しているか区別することができる。たとえば[HowPresent](#)属性に“[DateLine](#)”をセットすることによって[DateLine](#)要素に記述されている場所と関連があることを明示することができる。

```
<Location HowPresent="DateLine">
  <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>
```

ここで、[Location](#) 要素を複数繰り返し記述することにより、複数の場所の情報を示すことも可能である。

[TopicOccurrence](#) 要素は、[NewsComponent](#) 内にどのようなトピックが出現しているのかを表す。トピックの指定に使用する [Topic](#) 属性には、# 文字の後に現在の文章中にある [Topic](#) 要素の [Duid](#) 属性の値が続く [fragment identifier](#) を記述する。

[NewsML 仕様の変更点：NewsML v1.1]

DateLineDate 要素及び[Location](#) 要素は Version 1.1 で新規に追加された。

この例では、[ContentItem](#) は日本語で記述されていて、サブジェクトは IPTC Subject 「13008000」 (IPTC subject codes vocabulary では宇宙開発計画 “Space programme” と記述) であることがわかる。

```
<DescriptiveMetadata>
  <Language FormalName="ja"/>
  <SubjectCode>
    <SubjectMatter FormalName="13008000"/>
  </SubjectCode>
</DescriptiveMetadata>
```

[NskNewsML:1 記述形式]

DescriptiveMetadata 要素 ○
DescriptiveMetadata /Catalog 要素 ○
DescriptiveMetadata /Language 要素 ○
DescriptiveMetadata /Genre 要素 ×
DescriptiveMetadata /SubjectCode 要素 ○

DescriptiveMetadata /OfInterestTo 要素 …… ×
DescriptiveMetadata /TopicOccurrence 要素 … ×
DescriptiveMetadata /Property 要素 …… ×

Language 要素は、FormalName 属性で値を記述する。

(例) <Language FormalName="ja"/>

Subject、SbjectMatter、SubjectDetail、SubjectQualifier 要素の FormalName 属性の値で、IPTC のボキャブラリに不足するコードの追加に関しては、新聞協会を通じて IPTC に申請するものとする。

コラム：サブジェクトコードの管理方法について

サブジェクトコードはニュースの主題を示す情報で、言語、データの種類などに依存しないものとして、1997年にIPTCで作成され、メンテナンスされてきた。これまでも NewsML 以外に、IPTCの IIM、NITF などのニュースフォーマットにおいて世界中で使われてきた実績がある。

ニュースの主題を数値化、階層化することによりサブジェクトコードは表現される。英語の他、各国言語での説明を [SubjectCode](#) TopicSet ファイル中の [Description](#) 要素として追加することにより、どの言語でも使用できる。

サブジェクトコードはIPTCのカテゴリコード分科会でメンテナンスされ、2001年当時、以下のような手続きを必要とした。

- (1) 年3回開催されるIPTCの会議の3週間以上前にIPTCあてに提案を提出
- (2) カテゴリコード分科会での事前検討
- (3) IPTCの全体会議の分科会で検討、承認
- (4) IPTCの標準化委員会でどのように文書化するかを決定

2001年のIPTC年次総会で日本新聞協会から大幅な追加を要望し、過半数が承認された。追加手続きとしては、日本で追加を希望するサブジェクトコードを検討し英語の説明を付加したものが上記の流れに従って検討された。検討の中で、「スポーツリーグ名」「選挙情報」「政党」などの各国独自の情報を含むものは、キーワードに含むべきものとして承認されなかった。既存の米国の大リーグ名については過去に追加を認めたのは誤りと認識されたが、現時点では既に認められたものとして削除されていない。これらの国に依存する特殊な情報の表現方法はスポーツ、選挙など特殊なデータコンテンツ表現のためのマークアップ作成の中で検討される。国ごとの管理や部分的なローカルな使用など柔軟なサブジェクトコードの運用も提案されたが、世界での共通使用というサブジェクトコードの基本的な概念と相容れないものとして受け入れられなかった。また、この会議の中でメジャー、マイナーなどをサブジェクトクオリファイアに追加、一般的なプロスポーツリーグ情報を少し細分化して識別する方法がUPIから提案され一般的な方法として承認された。

追加が認められたサブジェクトコードと認められていないサブジェクトコードを以下に記述する。(日本語は暫定訳)

追加が認められたサブジェクトコード

漫画(Cartoon), アニメ(Animation), 検察(Prosecution), 組織犯罪(Organized Crime), 自動車事故(Road Accidents), 列車事故(Railway Accidents), 航空・宇宙事故(Air and Space Accidents), 海難事故(Maritime Accidents), 経済団体(Economic organizations), 入試(Entrance examination), 叙勲・賞(Award:Prize), 皇室(Imperial, Royal Matters), サミット(Summit), 少年問題(Juvenile delinquency), 核被爆者(Nuclear Radiation Victims), 軟式野球(Rubberball Baseball), 囲碁(Go)

追加が認められなかったサブジェクトコード

首長選挙(Parliamentary Elections), 議会選挙(Local Assembly Elections), 市長選挙(Gubernatorial, Mayoral Elections), 首相・大統領(Prime Minister, President), 諮問機関(Advisory panels), 与党(Ruling Parties), 野党(Opposition Parties), セントラルリーグ(Japanese Baseball-Central League), パシフィックリーグ(Japanese Baseball-Pacific League), J1リーグ(J-league first division), J2リーグ(J-league second division), パラリンピック(Paralympics)

2004年現在、この手順はSRS (Subject Reference System)への変更申請として管理されるようになった。この管理方法についてもIPTCで議論されており、詳細については、Subject Reference System Guidelinesに記述されている。これは、<http://www.iptc.org/> からたどることができる。主な内容は、申請が手順化されたこと、会議での議論が必要なものと、必要とせずに承認されるもの(ファストトラック)に分けられたことなどである。

2004年、追加が認められたサブジェクトコード

剣道(kendo), 銃剣道(jukendo), なぎなた(naginata), 弓道(kyudo), カバディ(kabaddi), セバタクロー(sepak takraw), 武術(wushu), ソフトテニス(soft tennis), 地価(land prices)

※ソフトテニス、地価は[SubjectDetail](#)へファストトラックとして申請した。

コラム：xml:lang vs Language

[xml:lang](#)属性はXML1.0で定義されており、その属性が含まれる要素の値を記述する言語を表現するためのもので、[XML](#)文書进行处理するためのメタデータとして位置付けられる。

対して、[Language](#)要素は、NewsMLの要素であり、[ContentItem](#)に対する言語という幾分概念的なメタデータとなっている。マルチ言語で表現されたニュース素材を扱う場合など、配信社が任意で設定可能である。

ここで言語の表現方法について記述するが、NewsML Version1.1 Functional Specificationで、「[xml:lang](#)属性の値は[IETF RFC3066](#)に従い記述しなければならない」と変更された（[5.4.2 SentFrom と SentTo](#)の項を参照）ことを受け、「同様に言語を表現する[Language](#)要素の値も[IETF RFC3066](#)である事が望ましい」との見解が示されている。ここで問題になるのが[IETF RFC3066](#)は、あらゆる言語コード（2文字）と国コード（2文字、または3文字）の組み合わせが理論上可能なため、これに対応した[TopicSet](#)の作成が困難なことである。[Language](#)要素による言語の表現は、[FormalName](#)属性による記述（[controlled vocabulary](#)で決まった値のみ許される）であるため、2003年のAGM会議において、「主要な20言語に対してまず[TopicSet](#)を作成し、その後リクエストに応じて更新する」ことが提案された。更には、「[IETF RFC3066](#)における規定は、[TopicSet](#)としては用意されていないが、十分にcontrollされたvocabularyと判断できる。そのため、[TopicSet](#)ファイルは不要である」という提案があった。

2004年現在、上記後者の意見が採用され、[Language](#)要素の値もRFC3066に順ずるとされており、[TopicSet](#)ファイルはない。

参考：「[5.4.2 SentFrom と SentTo](#)」

コラム：ニュースコンテンツの対象者指定

[OfInterestTo](#)要素は該当するNewsItemのコンテンツに対して、その内容が読者、視聴者に対して対象者の枠組みが定まっている場合に用いられる。[IPTC](#)の[TopicSet](#)には年齢世代別のターゲットが用意されている。その子要素の[Relevance](#)は、その対象者に対して、どの程度の関連性をもっているかを表す。コンテンツに対するコンテンツ対象者、さらにその対象者とコンテンツの関連性の強弱を表す。

5.10 NewsLines はメタデータの人間に対する局面を表す

[NewsComponent](#)は[NewsLines](#)を含むことができる。その目的はメタデータのある局面を人間が判読できる（出版できる）表現として与えること。[NewsLines](#)要素は、[HeadLine](#)、[SubHeadLine](#)、[ByLine](#)、[DateLine](#)、[CreditLine](#)、[CopyrightLine](#)、[RightsLine](#)、[SeriesLine](#)、[SlugLine](#)、[KeywordLine](#)下位要素を含む。[HeadLine](#)もあるときに限り[SubHeadLine](#)が現れてもよいという例外はあるが、これらすべてはオプションで繰り返し可能である。

[HeadLine](#)要素は表示可能な見出しを提供し、[SubHeadLine](#)要素は表示可能な補助見出しを提供する。

[ByLine](#)要素は著者／クリエイター情報の自然言語陳述を提供する。

[ByLineTitle](#)要素は自然言語で記述し、筆者や作成者の肩書きを表す。

[DateLine](#)要素は[NewsComponent](#)作成の日付および場所、またはそのいずれかの自然言語陳述を提供する。

[CreditLine](#)要素はクレジット情報の自然言語陳述を提供する。

[CopyrightLine](#)要素は著作権情報の自然言語陳述を提供する。

[RightsLine](#)要素は権利情報の表示可能な説明を提供する。これは著作権情報とは全く異なることに注意せよ。著作権情報は誰が[news object](#)の所有者であるかについての情報である。一方、権利情報は誰がどのような方法で、どのような状況のもとでそれを使用することが許されているかについてである。

[SeriesLine](#)要素は、続き物における[news object](#)の位置に関する表示可能な説明を提供する。

[SlugLine](#)要素はテキストの文字列を提供する。おそらくハイパーリンクおよびフォーマットिंगまたはいずれかに飾られており、NewsItemのslug lineを表示するのに使われる。("slug line"の用語の意味とその使い方は、個々の配信社が自身のワークフローと業務慣行の中で定義するものであることに注意する。)

[KeywordLine](#)要素は[news object](#)に関連した表示可能なキーワードのセットを提供する。これはNewsMLシステムが手動もしくは自動の検索を補助することに使える。

[NewsLine](#)要素はNewsMLの仕様に含まれない[newslines](#)の型を含むことを考慮に入れている。それぞれの[NewsLine](#)要素は1つの[NewsLineType](#)要素を含まなければならない。そして1つ以上の[NewsLineText](#)要素を含むことができる。1つ以上の[NewsLineText](#)要素があれば、それらが書かれた言語を表す[xml:lang](#)属性によって区別される。

[NewsLineType](#)要素はユーザー定義の[NewsLine](#)の型を表す。[FormalName](#)属性の値は[NewsLineType](#)の形式名である。その意味と許可される値は、[Vocabulary](#)属性と[Scheme](#)属性によって識別される[controlled vocabulary](#)によって決められる。

[NewsLineText](#)要素は、ユーザー定義型の[NewsLine](#)のテキストを含む。[NewsLineText](#)要素はプレーンテキストと[Origin](#)要素のどんな混合でも含むことができる。

[NewsLines](#)要素は、[NewsComponent](#)が持つすべての[NewsLine](#)のコンテナである。

```

<!ELEMENT NewsLines ((HeadLine , SubHeadLine* ) | (ByLine , ByLineTitle* ) | DateLine | CreditLine |
CopyrightLine | RightsLine | SeriesLine | SlugLine | KeywordLine | NewsLine)* >
<!ATTLIST NewsLines %localid; >

<!ELEMENT HeadLine (#PCDATA | Origin)*>
<!ATTLIST HeadLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT SubHeadLine (#PCDATA | Origin)*>
<!ATTLIST SubHeadLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT ByLine (#PCDATA | Origin)*>
<!ATTLIST ByLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT DateLine (#PCDATA | Origin)*>
<!ATTLIST DateLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT CreditLine (#PCDATA | Origin)*>
<!ATTLIST CreditLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT CopyrightLine (#PCDATA | Origin)*>
<!ATTLIST CopyrightLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT RightsLine (#PCDATA | Origin)*>
<!ATTLIST RightsLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT SeriesLine (#PCDATA | Origin)*>
<!ATTLIST SeriesLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT SlugLine (#PCDATA | Origin)*>
<!ATTLIST SlugLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT KeywordLine (#PCDATA | Origin)*>
<!ATTLIST KeywordLine %localid;
          xml:lang CDATA #IMPLIED >

<!ELEMENT NewsLine (NewsLineType , NewsLineText+)>
<!ATTLIST NewsLine %localid; >

<!ELEMENT NewsLineText (#PCDATA | Origin)*>
<!ATTLIST NewsLineText %localid;
          xml:lang CDATA #IMPLIED >

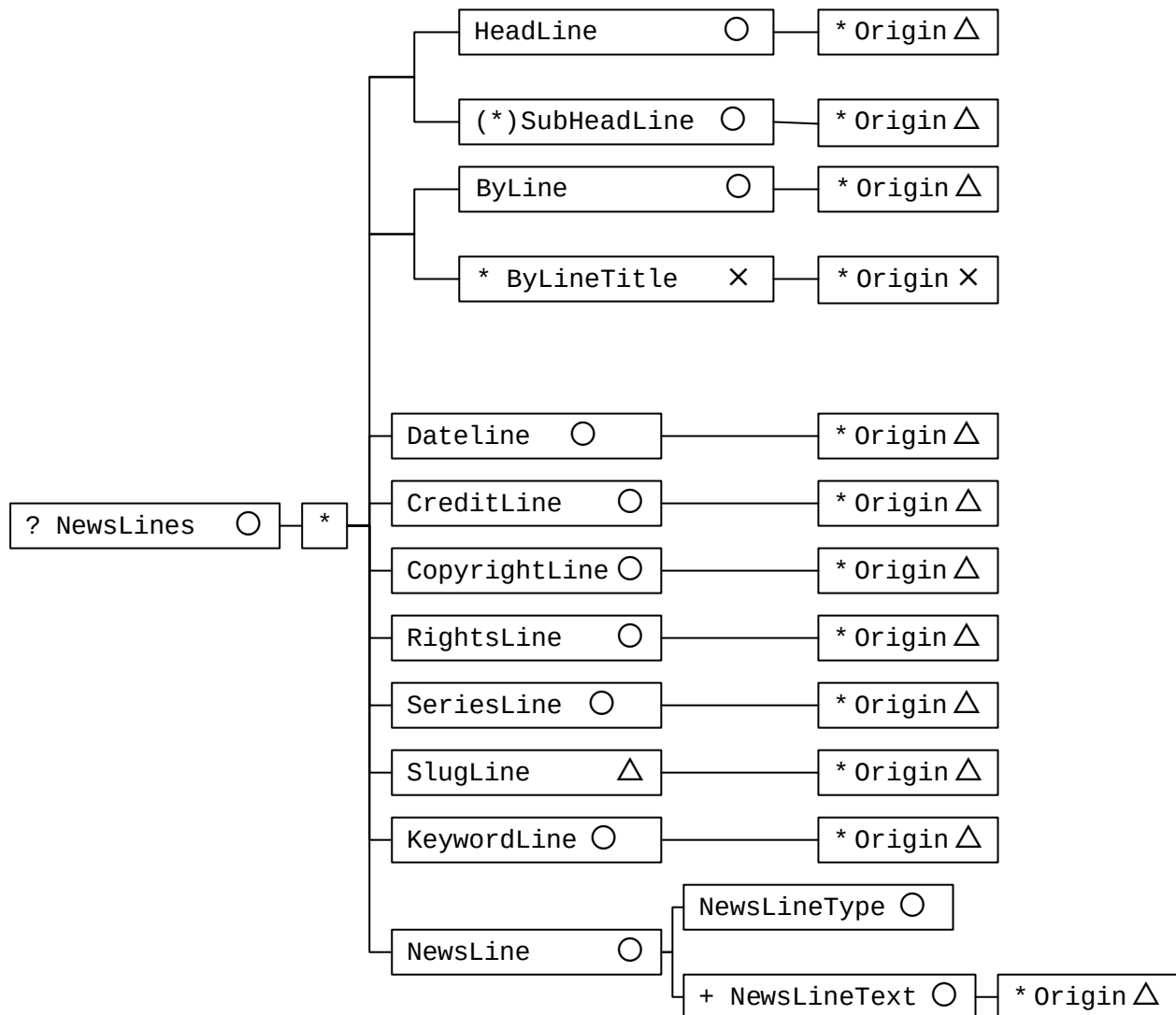
<!ELEMENT NewsLineType EMPTY>
<!ATTLIST NewsLineType %localid;
          %formalname; >

```

この例では、[Origin](#)要素はnews linesのパーツからローカルなトピックセットの中のトピックへリンクするのに使われている。この要素は当該のトピックについての外部情報源を参照する[Details](#)属性を持っている。付け加えると、ユーザー定義の[newslines](#) のタイプはローカルなトピックセットで宣言され、付加的な[NewsLine](#)要素で使われる。

```
<TopicSet Duid="LocalTopicSet" FormalName="person">
  <Topic Duid="person1" Details="http://mydomain.com/staff.xml#jwilson">
    <TopicType FormalName="person"
      Vocabulary="http://www.iptc.org/NewsML/topicsets/iptc-topictypes.xml"/>
  </Topic>
  <Topic Duid="position1" Details="http://mydomain.com/positions.xml#staffreporter">
    <TopicType FormalName="position" Vocabulary="#LocalTopicSet"/>
  </Topic>
  <Topic Duid="newspaper1" Details="http://mydomain.com/papers.xml#dailyrecord">
    <TopicType FormalName="newspaper" Vocabulary="#LocalTopicSet"/>
  </Topic>
  <Topic Duid="newslinetype1">
    <TopicType FormalName="NewsLineType"
      Vocabulary="http://www.iptc.org/NewsML/topicsets/iptc-topictypes.xml "/>
  <FormalName>ImpactLine</FormalName>
  <Description xml:lang="en-GB">An indication of the significance of the event
    described</Description>
  </Topic>
  <Topic Duid="topictype1">
    <TopicType FormalName="TopicType"
      Vocabulary="http://www.iptc.org/NewsML/topicsets/iptc-topictypes.xml"/>
    <FormalName>position</FormalName>
    <Description xml:lang="en-GB">An job function performed by a person.</Description>
  </Topic>
  <Topic Duid="topictype2">
    <TopicType FormalName="TopicType"
      Vocabulary="http://www.iptc.org/NewsML/topicsets/iptc-topictypes.xml"/>
    <FormalName>newspaper</FormalName>
    <Description xml:lang="en-GB">A publication that carries news.</Description>
  </Topic>
</TopicSet>
...
<NewsLines>
  <HeadLine>Clinton Addresses Crowd</HeadLine>
  <SubHeadLine>New policies announced</SubHeadLine>
  <ByLine>By <Origin Href="#person1">James Wilson</Origin></ByLine>
  <CreditLine><Origin Href="#position1">Staff Reporter</Origin> of <Origin
    Href="#newspaper1">The Daily Record</Origin></CreditLine>
  <NewsLine>
    <NewsLineType FormalName="ImpactLine" Vocabulary="#LocalTopicSet"/>
    <NewsLineText>Key pre-election rallying call</NewsLineText>
  </NewsLine>
</NewsLines>
```

[ツリー図]



[NSK解説]

[NewsLines](#)要素とその下位要素は、他のメタデータ用要素が機械的に処理できる厳密な形式での内容の記述を要求するのに対し、人間が読むことを想定した形式で値を記述するために用意されている。下位要素の内容は、必要ならば、読者・視聴者に対して生データとともに表示することができるものである。下位要素の[HeadLine](#)要素、[ByLine](#)要素、[DateLine](#)要素、[CreditLine](#)要素、[CopyrightLine](#)要素、[RightsLine](#)要素、[SeriesLine](#)要素、[SlugLine](#)要素、[KeywordLine](#)要素および[NewsLine](#)要素の記述順序、記述回数は任意である。

[SubHeadLine](#)要素は[HeadLine](#)要素を記述した場合に、[ByLineTitle](#)要素は[ByLine](#)要素を記述した場合にのみ記述することができる。記述言語を変えて複数の[SubHeadLine](#)要素、[ByLineTitle](#)要素を記述することもできる。

[CreditLine](#)は、[news object](#)に書き添える提供社の名前を書く。

[NewsML仕様の変更点：NewsML v1.1]

- [ByLineTitle](#)要素が追加された。筆者や作成者の肩書きをあらわすために使用する要素で、記述回数は任意である。[ByLine](#)要素を記述した直後にのみ記述することができる。
 - [NewsLines](#)要素の下位要素（[HeadLine](#)要素、[ByLine](#)要素、[DateLine](#)要素、[CreditLine](#)要素、[CopyrightLine](#)要素、[RightsLine](#)要素、[SeriesLine](#)要素、[SlugLine](#)要素、[KeywordLine](#)要素、[NewsLine](#)要素）の記述順序が任意となった（NewsMLv1.0では下位要素の記述順序が規定されていた）。
 - [SubHeadLine](#)要素の記述回数が任意になった。Version1.0では0回または1回と規定されていたもの。
- （各下位要素の説明）

要素名	説明	例
HeadLine	見出し	わかり易いNewsML
SubHeadLine	副見出し	初心者向け解説
ByLine	作成者	人の名前
DateLine	作成日	2001年 5月
CreditLine	news object に添える提供社の名前	NSK
CopyrightLine	コピーライトの表記	2001 NSK All rights reserved
RightsLine	権利情報	協会加盟社だけ利用可能
SeriesLine	シリーズものの名称と回数	連載 1 回目
KeywordLine	検索用のキーワード	NewsML、解説
NewsLine	その他、上で表現できない情報領域	※「絵解き」や「編注」の情報

フォーマットサンプル

```
<NewsLines>
  <!--以下のXXXXXXLinesは、省略可能-->
  <HeadLine xml:lang="en-GB">Simple NewsML</HeadLine>
  <SubHeadLine xml:lang="en-GB">for beginner</SubHeadLine>
  <HeadLine xml:lang="ja-JP">わかり易いNewsML</HeadLine>
  <SubHeadLine>初心者向け解説</SubHeadLine>
  <ByLine>人の名前</ByLine>
  <DateLine>2001年 5月</DateLine>
  <CreditLine>NSK</CreditLine>
  <CopyrightLine xml:lang="en-GB">2001 NSK ALL rights reserved</CopyrightLine>
  <RightsLine>協会加盟社だけ利用可能</RightsLine>
  <SeriesLine>連載 1 回目</SeriesLine>
  <KeywordLine>NewsML</KeywordLine>
  <KeywordLine>解説</KeywordLine>
  <NewsLine>
    <NewsLineType FormalName="Caption"/>
    <NewsLineText>NewsLinesの説明</NewsLineText>
  </NewsLine>
</NewsLines>
```

補) 上記サンプルでの[RightsMetadata](#)は以下のようになると考えられる：

```
<RightsMetadata>
  <Copyright>
    <CopyrightHolder>日本新聞協会</CopyrightHolder>
    <CopyrightDate>2001</CopyrightDate>
  </Copyright>
</RightsMetadata>
```


[NskNewsML:1記述形式]

NewsLines要素	○
NewsLines/HeadLine要素	○
NewsLines/SubHeadLine要素	○
NewsLines/ByLine要素	○
NewsLines/DateLine要素	○
NewsLines/CreditLine要素	○
NewsLines/CopyrightLine要素	○
NewsLines/RightsLine要素	○
NewsLines/SeriesLine要素	○
NewsLines/KeywordLine要素	○
NewsLines/NewsLine要素	○

複数言語が混在する場合は、**xml:lang** を使い言語を切り換える方法で、複数の要素を並列表記する。テキスト部分はすべて自由形式だが、**NewsLineType** の **FormalName** は NSK オリジナルボキャブラリを使用する。

[NskNewsML:1.2記述形式]

NewsLines要素	○
NewsLines/HeadLine要素	○
NewsLines/SubHeadLine要素	○
NewsLines/ByLine要素	○
NewsLines/ByLineTitle要素	×
NewsLines/DateLine要素	○
NewsLines/CreditLine要素	○
NewsLines/CopyrightLine要素	○
NewsLines/RightsLine要素	○
NewsLines/SeriesLine要素	○
NewsLines/KeywordLine要素	○
NewsLines/NewsLine要素	○

複数言語が混在する場合は、**xml:lang** を使い言語を切り換える方法で、複数の要素を並列表記する。テキスト部分はすべて自由形式だが、**NewsLineType** の **FormalName** は NSK オリジナルボキャブラリを使用する。

NskNewsML:1.2では、NskNewsML:1との互換性を保つために、NewsML v1.1の変更を適用してはならない。

5.11 NewsItems の改版

NewsItem の改版は、改版される前の NewsItem と同じ [ProviderId](#)、[DateId](#)、[NewsItemId](#) を持つ NewsItem を含む新しい NewsML 文書を作成し発行することである。

NewsItem のうち、[NewsManagement](#) および／または [Identification](#) の 1 つ以上の下位要素を変更（NewsItem のほかの部分の更新を行わない）するために、[RevisionId](#) 要素は、オリジナルのものとまったく同じであるべきで、その際 [Update](#) 属性の値は「A」を設定する。そして、この NewsItem には、あらゆる変更を含んだ完全な [Identification](#) および [NewsManagement](#) 要素を入れるべきで、ほかには何も入れてはならない。

もし NewsItem の [NewsManagement](#) / [Identification](#) 以外の部分を何らかの方法で変更する場合、[RevisionId](#) には、以前のものより大きい数字をセットする。さらに、[PreviousRevision](#) 属性には、その前の版の [RevisionId](#) と同じ値をセットする。その時、下記の 2 つの選択がある：

- なされた変更を組み入れた完全な NewsItem を発行する。そして [RevisionId](#) 要素の [Update](#) 属性の値を「N」に設定する。
- NewsItem の [NewsComponent](#) 下位要素を、新しいドキュメントに含めず、その代わりに変更したことを示す 1 つ以上の [Update](#) 要素を提供する。そして [RevisionId](#) 要素の [Update](#) 属性の値を「U」に設定する。

[Update](#) 要素は、既存の NewsItem への加えた変更を示す。これには、挿入(insertion)、置き換え(replacement)、あるいは削除(deletion)がある。ただし、[Update](#) 要素は [NewsManagement](#)、[Identification](#) 要素、あるいはそれらの子要素(descendants)のどれかを変更するという目的では使うことができないことに注意する。NewsItem のこれらの部分への変更は、[Identification](#) および [NewsManagement](#) 要素だけを、現在の版番号で NewsItem を発行することにより行う。これにより、以前の [Identification](#) および [NewsManagement](#) 要素が受信側のシステムで置き換える。[Update](#) 要素は、以下の種類の下位要素をいくつか含む：

- [Delete](#)（削除）
- [Replace](#)（置き換え）
- [InsertBefore](#)（前に挿入）
- [InsertAfter](#)（後ろに挿入）

NewsItem の新しいコピーを作成するのは、受信側のシステムである。既にあるか、配信側が要求した NewsItem の前の版に対して [Update](#) 指示に従う。各 [Update](#) 要素のそれぞれの下位要素を、変更が生じた順に、順番に処理し NewsItem の新しい版を作成する。各下位要素の [DuidRef](#) 属性の値は、前の版で持っていた要素の [Duid](#) に合わせる必要がある。これは指示に合った要素である。削除（Delete）の場合、該当する要素は改訂された NewsItem から削除する。置き換え（[Replace](#)）の場合、該当する要素は [Replace](#) 要素の内容に置き換える。[InsertBefore](#) の場合、[InsertBefore](#) 要素の内容が、新しい版で該当する要素の前に付け加える。[InsertAfter](#) の場合、[InsertAfter](#) 要素の内容が、新しい版で該当する要素の後に加える。

```

<!ELEMENT Update (InsertBefore | InsertAfter | Replace | Delete )*>
<!ATTLIST Update %localid; >

<!ELEMENT InsertBefore ANY >
<!ATTLIST InsertBefore %localid;
                DuidRef CDATA #REQUIRED >

<!ELEMENT InsertAfter ANY >
<!ATTLIST InsertAfter %localid;
                DuidRef CDATA #REQUIRED >

<!ELEMENT Replace ANY >
<!ATTLIST Replace %localid;
                DuidRef CDATA #REQUIRED >

<!ELEMENT Delete EMPTY >
<!ATTLIST Delete %localid;
                DuidRef CDATA #REQUIRED >

```

[NSK 解説]

◆Update要素◆

[Update](#)要素は、以前の版の [NewsItem](#) に対する変更を示す。このとき、[RevisionId](#)要素の [PreviousRevision](#)属性値はその前の版の [RevisionId](#)要素値と同じ値とし、[Update](#)属性の値を「U」に設定する。[Update](#)要素は[NewsManagement](#)や[Identification](#)に関する変更には使用しないことに注意する。

変更指示の種類を

- [Delete](#) (削除)
- [Replace](#) (置き換え)
- [InsertBefore](#) (前に挿入)
- [InsertAfter](#) (後ろに挿入)

の下位要素で記述する。変更対象場所の指定は、各下位要素の [DuidRef](#)属性に前の版の [NewsItem](#) 内の変更対象要素の [Duid](#)値を設定することで行う。

コラム：Update要素の実装について

各下位要素について DTD や Schema の定義ではどのような形を取ることもできるが、変更指示反映後に生成される新しい版の NewsItem が NewsML 的にエラーとならない必要がある。

Update の情報を取り込み新しい NewsItem の完成版を作成するのは、受信社側のシステムとなる。一方で NewsItem のバージョン情報を設定するのは送信社側のシステムとなる。ここで注意が必要なのは、以前のバージョンの NewsItem が受信社側になく、受信社側で新しい NewsItem の完成版を作成するのが不可能であるということである。特に受信社が送信社にアクセスして NewsML を取り出すようなシステムではこのような問題が起こりやすいだろう。

Update の実装には送信側および受信側のシステム側で厳密な NewsItem のバージョン管理がなされていること、Update で関連する NewsItem について、送受信双方のシステムで同じように存在していることが必要となる。また、データ交換を行う当事社のシステムの違いにより、完成版の NewsItem に差異が生じないか注意する必要がある。

なお、Update 要素は NskNewsML:1.2 では使用停止としている。

◆NewsItemの変更◆

NewsML では、発行済みの NewsItem を変更する手段を提供している。変更には次の 3 種類がある。

- 1) NewsItem 全体を、すべて送りなおす場合(改版される)
- 2) NewsItem/NewsManagement の変更をする場合(改版されない)
- 3) NewsItem/Update 要素を使って NewsItem 内の部分または全体の変更指示を送る場合(改版される)

以下に詳細を説明する。

- 1) NewsItem 全体を、すべて送りなおす場合(改版される)

これは、NewsComponent をはじめ、すべての内容を含めた NewsItem を、新しい版として送ることである。

この場合、NewsItem/Identification/NewsIdentifier/RevisionId 要素を次のように記述し、以下の条件を満たさなければならない。

```
<RevisionId PreviousRevision="1" Update="N">2</RevisionId>
```

- Update 属性値は"N"でなければならない。
- PreviousRevision 属性値(上記の"1")は、NewsItem の前の版と同じでなければならない。
- RevisionId 要素の内容(上記の 2)は、前の版よりも大きい数字にしなければならない。

上記に伴い、PublicIdentifier の値も変化するので注意する必要がある。具体的には、urn:newsml:{ProviderId}:{DateId}:{NewsItemId}:{RevisionId}{RevisionId@Update} の {RevisionId}{RevisionId@Update} が変化する。

(例)

```
<PublicIdentifier>urn:newsml:provider.or.jp:20010503:T001:2</PublicIdentifier>
```

(例 1)

--発行済みの版--

```
<NewsIdentifier>
  ...
  <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
  <PublicIdentifier>
    urn:newsml:provider.or.jp:20010503:T001:1
  </PublicIdentifier>
</NewsIdentifier>
```

...

--新しい版を送り、改版する--

```
<NewsItem>
  <Identification>
    <NewsIdentifier>
      ...
      <RevisionId PreviousRevision="1" Update="N">2</RevisionId>
      <PublicIdentifier>
        urn:newsml:provider.or.jp:20010503:T001:2
      </PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    ...
  </NewsManagement>
  <NewsComponent>
    <!-- 新しいニュースコンポーネント -->
    ...
  </NewsComponent>
</NewsItem>
```

2) NewsItem/NewsManagement の変更をする場合(改版されない)

これは既に発行済みのNewsItemに対して、NewsItem/IdentificationやNewsItem/NewsManagementを変更し、NewsItemの他の部分に何の変更も行わない場合に使用する。

この場合、NewsItem/Identification/NewsIdentifier/RevisionId 要素を次のように記述し、以下の条件を満たさなければならない。

```
<RevisionId PreviousRevision="0" Update="A">1</RevisionId>
```

- [Update](#)属性値は"A"でなければならない。
- [PreviousRevision](#)属性値(上記の"0")と[RevisionId](#)要素の内容(上記の 1)は、変更される NewsItem の版と同じでなければならない。
- NewsItem には変更部分を含む完全な形の[Identification](#)と[NewsManagement](#)要素のみを記述し、NewsItem の他の要素を記述してはならない。

上記に伴い、[PublicIdentifier](#)の値も変化するので注意する必要がある。具体的には、urn:newsml:{ProviderId}:{DateId}:{NewsItemId}:{RevisionId}{RevisionId@Update}の{RevisionId@Update}が省略されず、Aになり、{RevisionId}A の記述になる。

受信側は、修正の反映時にもとの[NewsIdentifier](#)を変更してはならない。

(例2) 発行済みの NewsItem のステータスを変更する場合

--発行済みの版--

```
<NewsIdentifier>
...
<RevisionId PreviousRevision="0" Update="N">1</RevisionId>
<PublicIdentifier>
urn:newsml:provider.or.jp:20010503:T001:1
</PublicIdentifier>
</NewsIdentifier>
```

--上記の版を公開待機にする場合--

```
<NewsItem>
<Identification>
<NewsIdentifier>
...
<RevisionId PreviousRevision="0" Update="A">1</RevisionId>
<PublicIdentifier>
urn:newsml:provider.or.jp:20010503:T001:1A
</PublicIdentifier>
</NewsIdentifier>
</Identification>
<NewsManagement>
...<!--すべて置き換えられるので、省略せずに記述する。-->
<Status FormalName="Embargoed"/>
...
</NewsManagement>
<!-- NewsComponent は記述してはいけない -->
</NewsItem>
```

上記を受けた以降、urn:newsml:provider.or.jp:20010503:T001:1 で指定される NewsItem の内容は、ステータスが公開待機(Embargoed)になっていなければならない。

(例3) 発行済みの NewsItem へ関連情報(NewsManagement/AssociatedWith)を追加する場合

--発行済みの版--

```
<NewsIdentifier>
...
<RevisionId PreviousRevision="0" Update="N">1</RevisionId>
<PublicIdentifier>
urn:newsml:provider.or.jp:20010503:T001:1
</PublicIdentifier>
</NewsIdentifier>
```

--上記の版へ関連情報(NewsManagement/AssociatedWith)を追加する場合--

```
<NewsItem>
<Identification>
<NewsIdentifier>
...
<RevisionId PreviousRevision="0" Update="A">1</RevisionId>
<PublicIdentifier>
urn:newsml:provider.or.jp:20010503:T001:1A
</PublicIdentifier>
</NewsIdentifier>
</Identification>
<NewsManagement>
...<!--すべて置き換えられるので、省略せずに記述する。-->
<AssociatedWith NewsItem="urn:newsml:provider.or.jp:20010503:T029">
```

```

    <Comment>解説記事</Comment>
  </AssociatedWith>
  ...
</NewsManagement>
<!-- NewsComponent は記述してはいけない -->
</NewItem>

```

3) NewsItem/Update要素を使って NewsItem 内の部分または全体の変更指示を送る場合(改版される)

これは[Update](#)要素を使用して、主に[NewsComponent](#)以下の変更を指示する。

[Update](#)要素を使用して、主に[NewsComponent](#)以下の内容の一部または全部の変更を指示する。

この場合、NewsItem/Identification/NewsIdentifier/RevisionId 要素を次のように記述し、以下の条件を満たさなければならない。

```
<RevisionId PreviousRevision="1" Update="U">2</RevisionId>
```

- [Update](#)属性値は"U"でなければならない。
- [PreviousRevision](#)属性値(上記の"1")は、NewsItem の前の版と同じでなければならない。
- [RevisionId](#)要素の内容(上記の 2)は、前の版よりも大きい数字にしなければならない。

上記に伴い、[PublicIdentifier](#)の値も変化するので注意する必要がある。具体的には、urn:newsml:{ProviderId}:{DateId}:{NewsItemId}:{RevisionId}{RevisionId@Update}の{RevisionId@Update}が 省略されず、U になり、{RevisionId}U の記述になる。

(例4)発行済みの NewsItem の一部を、Update を用いて置き換える場合

--発行済みの版--

```

<NewItem>
  <Identification>
    <NewsIdentifier>
      ...
      <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
      <PublicIdentifier>
        urn:newsml:provider.or.jp:20010503:T001:1
      </PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    ...
  </NewsManagement>
  <NewsComponent>
    ...
    <!-- NewsComponent の二階層記述... -->
    <NewsComponent Duid="Component-1">
      ...
    </NewsComponent>
    <NewsComponent Duid="Component-2">
      ...
    </NewsComponent>
    ...
  </NewsComponent>
</NewItem>

```

--新しい版を送り、改版する--

```
<NewsItem>
  <Identification>
    <NewsIdentifier>
      ...
      <RevisionId PreviousRevision="1" Update="U">2</RevisionId>
      <PublicIdentifier>
        urn:newsml:provider.or.jp:20010503:T001:2U
      </PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    ...
  </NewsManagement>
  <!-- ニュースコンポーネントの一部を置き換え -->
  <Update>
    <Replace DuidRef="Component-1">
      <NewsComponent Duid="Component-1U">
        ...
      </NewsComponent>
    </Replace>
  </Update>
  ...
</NewsItem>
```

5.12 ポインタの使用

NewsML は、データを内部に含むよりもむしろ外部に置かれたデータを参照するため、ポインタの使用をサポートしている。この手法は [ContentItem](#) 内にある外部データ参照の際に使われる。また新しい文書にすべての内容をコピーする必要がなく、NewsML 中の既存の [NewsItem](#) を含む際に使われることもある。

[ContentItem](#) の場合、[ContentItem](#) の [Href](#) 属性は、[5.8 ContentItem の構造](#) でも説明したように、外部オブジェクトを参照するために使用する。

[NewsItem](#) の場合、[NewsItemRef](#) は、[NewsItemRef](#) 要素を置き換えると判断する [NewsItem](#) に対するポインタであり、[NewsItemRef](#) は [NewsItem](#) に置き換えて解釈する。[NewsItem](#) 属性は関連した [NewsItem](#) へのポインタである。その値は http URL または NewsML URN か、現在の文書中で # 記号の後に [NewsItem](#) の [Duid](#) を置いて構成している [fragment identifier](#) である。オプションの [Comment](#) は、この [NewsItem](#) を含む理由を解説するために使うことができる。

```
<!ELEMENT NewsItemRef (Comment*)>
<!ATTLIST NewsItemRef %localid;
                    NewsItem CDATA #IMPLIED >
```

5.13 NewsML の発展

NewsML は、それぞれが宣言された [naming scheme](#) によって決められている名前を持つ、一般的な [Metadata](#)、[Property](#)、[Label](#)、[NewsLine](#) 要素を提供する。これらの要素は、コントロールされた方法で、新たなメタデータや [newsline](#)、[label](#) を加えて使うことができる。これは、NewsML 文書に繰り返し拡張する表現能力があることを意図している。NewsML 自体の新しい版が公開されるとき、NewsML DTD や [schema](#) に、新たなメタデータや [newsline](#)、[label](#) のいくつか、あるいは全部を付け加えることが可能である。

5.14 認証とセキュリティ

[AdministrativeMetadata](#) は、[NewsComponent](#) のソース(著作者、発行者、再頒布者など)を特定する。このため、NewsML 文書の受信者は、それが誰から、または、どんな組織から来たものかを特定でき、信頼できる情報かどうか判断することができる。

NewsML では認証方法や [news object](#) のデジタル署名付与の仕組みを明示的に準備していない。[W3C](#) の XML-Signatures 仕様が [W3C](#) 勧告となり、そこで定義された仕組みが使われるようになるのを待つことになる。

6 用語

AdministrativeMetadata	管理メタデータ。 XML コンテンツ管理システムにおいて、 NewsComponent の出所についての情報と、名付け方の指示を提供するメタデータ。
AllowedValues	プロパティに対して許される値の範囲を決める controlled vocabulary を示す Property 要素の属性。
AssignedBy	だれが、あるいはどのシステムが、メタデータを割り当てた(assigned)のかを示す。
assignment	要素からなるエンティティ。その要素により、メタデータを割り当て(assigned)た人物またはシステム、信頼性の度合い、アサインメントに与える重要性、文脈(context)での参照トピックの存在という性質、などを判断できる。
AssociatedWith	関連した素材を伴った NewsItem への参照。
attribute (属性)	XML の構成は、 XML 要素のプロパティを表すペアの name-value から成り立っている。属性宣言は要素の開始タグに含まれている。 例: <pre><MyElement MyProperty="myvalue"/></pre> MyElement要素は、値が myvalue の MyProperty プロパティを持っている。
BasisForChoice	NewsComponent の下位要素。その content は、 NewsComponent 内の各アイテムに関連し、アイテム間の選択基準として使われている値を持つデータオブジェクトを識別する XPath 宣言である。
ByLine	著者/作者の情報を示している。
ByLineTitle	表示可能な筆者や作成者の肩書きの情報。
Catalog	Resource要素とTopicUse要素の親要素(container)。Resource要素はURNs からURLs へのマップをし、すぐ上のCatalog要素から始まるサブツリー内のある要素の形式名に適用されるdefault vocabulariesを示している。
Characteristics	解釈前あるいは解釈後のデータを処理するために必要なシステム要件に関係のある ContentItem の物理的な特徴についての情報を提供する。これは、ファイルサイズのバイト数といったものや、ユーザーがcontrolled vocabulariesを介して定義したり、先々のバージョンでNewsML DTDに加えられるかもしれない他のプロパティをカバーする。
Comment	さまざまな言語により、現行の要素を記述したり、その要素について記述すること。commentの親要素に含まれる情報を拡張した、人間が理解できる形の付加情報を提供する。
Complements (補完)	必要とされているフル情報の一部しか提供されていないとき、付け加えるべきNews objects。
Confidence	所定の文脈(context)で割り当てられた topic reference の信頼度。Confidence属性の値は controlled vocabulary によって制御されている。
content	コンテンツ。内容。要素の開始タグと終了タグの間にあるすべてのデータ。 Example:

```
<MyElement>text<ContentItem Href="a.xml"/></MyElement>
```

MyElement要素の[content](#)は次の行である。

```
text<ContentItem Href="a.xml"/>
```

[ContentItem](#)要素はcontentを持っていない。

ContentItem	人間に対しプレゼンテーションすることを意図した表現コンテンツ（たとえばテキスト、イメージ、ビデオ、オーディオなど）を伝達するデータオブジェクトを含み、そのデータオブジェクトへのポインタを提供する news object 。
Context	その値がXPathパターンであるTopicUseの属性。現在の Catalog に当てはまるサブツリー内にあり、参照トピックが使用された context を示す。
Contribution	ニュースオブジェクトの制作や変更を行うことを目的とするParty (NewsItem に対し特別な関係がある個人、会社、組織) の関与内容
Contributor	生成されたニュースオブジェクトの修正や拡張を行った個人、会社、組織あるいはその組み合わせを示す。
controlled vocabulary	定義された用語とその意味のリスト。公式変更手順により保守される。(naming scheme参照)。
Copyright	news object の著作権。
CopyrightDate	著作権日付の口語表現。
CopyrightHolder	著作権保持者情報の口語表現。
CopyrightLine	著作権情報の口語表現。
Creator	news object を作成した個人、会社、組織またはその組み合わせ。
CreditLine	信用情報の口語表現。
DataContent	ContentItem の content を保持するデータ。
DateAndTime	日付とオプションの時刻またはその双方の形式表現。ISO 8601 基本フォーマット (CCYYMMDDTHHMMSS {+ または -} HHMM) (世紀、年、月、日、時刻セパレータ、時、分、秒、タイムゾーンセパレータ、時、分) で表現される。システムで自動処理可能。
DateId	NewsItem の日付識別子。ISO 8601 の簡略日付フォーマット(CCYYMMDD)。DateIdは NewsItem の形式的な識別子であり、同じ NewsItem の一連のリビジョンでは同じ値が維持されなければならない。
DateLabel	日付および（または）時刻の文字列表現。人が NewsItem を確認するために使用。
DateLine	作成された日付、場所の口語表現。
declaration (宣言)	DTDの中で、そのDTDに沿ったドキュメントの特定の構造を定義する文字列。
default vocabulary	他の controlled vocabulary を特に参照し、上書きされるまで、あるいは上書きしない場合、デフォルトの意味と認められた値を与える controlled vocabulary 。

DefaultVocabularyFor	default vocabulary を与える親リソースを示す。 default vocabulary は NewsML ドキュメントのサブツリーの特定のパートに出現し、意味と許可されたデータ値を決定する。
Delete	現在の NewsItem の前の版の NewsItem 中の指定された要素の削除命令。
DerivedFrom	NewsItem がどこから得られたかを示す。
Description	Topic を示す記述。それによって、 Topic に関連付けられた形式名の意味を示す。オプションの Variant 属性により同じ言語で書かれた複数の記述が可能のため、他のものと区別できる。
DescriptiveMetadata	NewsComponent の内容を記述するメタデータ情報。
Details	Topic 要素の属性。URL またはURNの形式で中で Topic に関する追加情報へのポインタを提供。
DOCTYPE declaration	XML ドキュメント中の特別な declaration 。そのドキュメントが従う DTD を含む外部ファイルを指定する。
DTD	Document Type Definition（文書型定義）。 XML ドキュメントの構造を決定するdeclarationsのセット。 DTD はドキュメント自体の中にある 内部サブセット (internal subset) 、ドキュメントのDOCTYPE declaration により参照されるファイル内にある 外部サブセット (external subset) 、または両者のその組み合わせに含むことができる。
Duid	ドキュメントユニークな識別子 (document-unique identifier)。このオプションの属性により、要素は NewsML ドキュメントの中で唯一のものとして識別される。
DuidRef	参照要素 の Duid 属性に値が一致する属性。
element (要素)	XML 文書の構成要素(component)。要素は element type の名前を含む開始タグとオプションのいくつかの属性から始まる。さらに、（下位要素と呼ばれる）別の要素からなるいくつかの content 、テキスト、またはその双方を含むことができる。要素は終了タグで終わるか、 content がない時は開始タグの最後にスラッシュが付加される。
Example: <pre><MyElement>some text<EmptyElement/></MyElement></pre> これは text と EmptyElement 型の空要素を含む MyElement 型の要素。	
element type	XML 要素のカテゴリ。開始、終了タグにおける名称で識別される。特定のelement type の要素は DTD もしくは schema 中の element type のdeclarationsで定義された構造化規則を満たさなければならない。
encoding (エンコーディング)	データ・オブジェクトに含まれるデータを解釈する時に適用される規則。encoding の例としては、ASCII、UTF-8、UTF-16、base64、uuencode、zip など。 XML ファイルは、これらの encodings(それらはバイトストリームを文字ストリームに翻訳することができる規則を決定する)の内のいずれかを使用することができる。
Encoding	ContentItem の内容を包含するデータの encoding。
EndDate	使用権が終了する日付を指定する口語表現。

entity (エンティティ)	XML 文書に参照によって含むことができるデータ・オブジェクト。エンティティは文字番号によって参照された文字列、 DTD か schema の中の宣言で定義されたテキスト、あるいはテキストか他の種類のデータ(それらはオーディオ、ビデオ、イメージのようなバイナリーデータを含んでいるかもしれない)のいずれかを含んでいる外部ファイルであり得る。
entity reference	エンティティに対してポインタとなっている XML 文書の中の文字列で文書内にある。例えば、“The NewsML functional specification”が nfs という名前のエンティティとして定義された場合、“Please refer to the &nfs; for details”というフレーズでは、“&nfs”はentity referenceであり、そのフレーズは“Please refer to the NewsML functional specification for details”という意味となる。
equivalents (等価)	含んでいる情報が等価であるので、選択が行なわれるべきNews objects。
EquivalentsList	NewsComponent の属性の1つで、その中の news objects が、内容と／または意味において別の news objects に対して equivalents (等価) であるかどうか、「あるいは complements (補完) であるかどうか、を指し示すもの。
Essential	NewsComponent の属性の1つで、配信社がこの NewsComponent が重要であるとみなしているかどうかを示す。
Euid	要素を唯一（ユニーク）なものにする識別子（element-unique identifier）。すべての NewsML element type のオプション属性。同じ親要素内の同じ element type の間で、要素が唯一のものとして（ユニークに）識別されることを可能にする。
external subset	外部サブセット。 XML 文書の構造を管理し、その文書の DOCTYPE declaration から参照された DTD に含まれるdeclarations。
FileName	NewsItem の、示唆されたかあるいは実際の記憶ファイル名。
FirstCreated	NewsItem が最初に作成された日付。オプションとしてその時間。ISO8601 基本フォーマットで表現する。
formalname	FormalName 、 Vocabulary 属性、 Scheme 属性からなるエンティティ。 FormalName は、 controlled vocabulary によってその意味が決定される文字列からなる。 Vocabulary 属性があれば、それは FormalName の意味を解くために使用することができる controlled vocabulary である TopicSet へのポインタを供給する。 Scheme 属性があれば、それは controlled vocabulary 中に多分多数ある naming schemes のうち、 FormalName を管理しているものを識別するのに役立つ。
FormalName	controlled vocabulary の中の naming scheme によって意味が決定される文字列。 controlled vocabulary は(要求はされないが)、NewsML TopicSet の形式をとる場合もある。
format(フォーマット)	データオブジェクト内の情報を運ぶために使われるファイルタイプ。フォーマットにより、オブジェクトを加工したり、翻訳、表現したりすることが可能なアプリケーションが決まる。format の例として、GIF、JPEG、WAV、Microsoft Word、 XML などがある。
Format	ContentItem のフォーマットの表示。
fragment identifier	URL または URN のうちでリソース内の位置またはサブストリングを識別する部分。#文字によって、 URL あるいは URN の主要部分から分けられる。
FutureStatus	指定された将来の日付における NewsItem のステータスの表示。

Genre	NewsComponent のジャンルの表示。
Geography	特定の使用権が当てはまることを指定された地域の口語での表現。
HeadLine	表示することができるヘッドライン。
HowPresent	メタデータを適用する方法の表現
Href	情報が NewsML 文書または外部リソースのどこにあるかを指し示す属性。
Identification	NewsItem の識別に役立つメタデータ。 NewsIdentifier 、任意の NameLabel 、 DateLabel およびオプションで反復可能な Label を含む。
IETF	Internet Engineering Task Force
Importance	メタデータを付与する party がつけた重要性のランクづけ。
inclusion by reference	参照による内包。対象物自体の代わりに、データ対象物へのポインタを文書内で使用すること。この機構により、ほんのわずかの文字を伝えることによって、大きい NewsML 文書を送ることを可能になる。伝えられた文字のいくつかがポインタになる。そしてそれは、NewsML 文書が解釈されるか使われるときに、対象物自体に置き換えられる。
InsertAfter	NewsItem の内部で、指定された要素の後に content を挿入する指示。
InsertBefore	NewsItem の内部で、指定された要素の前に content を挿入する指示。
Instruction	ニュース配信社から NewsItem 受信社への指示。
internal subset	内部サブセット。文書構造を定義する declarations の一部あるいはすべてを含んでいる XML 文書の一部分。内部サブセットにない declarations は、 外部サブセット (external subset) にある。
IPTC	International Press Telecommunications Council (国際新聞電気通信評議会)
KeywordLine	news objects に関係のあるキーワードの表示可能な一群。これは、手動、あるいは自動化検索を助けるために NewsML システムにより使用することができる。
Label	NewsItem 用の人が解読可能なラベル。
LabelText	特定の LabelType の Label を構成するテキスト。
LabelType	Label のユーザー定義の型。 FormalName 属性の値は、LabelType に対しての形式名である。
Language	ContentItem で使用される用語または言語の識別子。
Location	NewsItem の内容に有意な、実際の場所を表す識別子。
Limitations	特定の使用権に適用される用語と条件の自然言語表現法。
media type	データ対象物に含まれる情報を人に示す媒体の種類。 media type の例は、ビデオ、オーディオ、ラスター・イメージ、ベクトル・グラフィック、テキストなどである。
MediaType	ContentItem のメディアタイプの指示。

metadata (メタデータ)	システムがデータ対象物を適切に処理することを可能する目的で、データ対象物と結びつけられたデータ。システムは、コンピュータ・アプリケーション、人間が処理するビジネス・プロセス、またはその2つの組み合わせがある。
Metadata	metadata のユーザー定義型のための入れ物。
MetadataType	この Metadata 要素内の Property 要素により表されるメタデータ型の指示。
MIME	マルチパート・インターネット・メール・エクステンション。 IETF の公式の仕様。インターネット上で伝達されるデータ対象物を解釈、処理または表現する能力のあるアプリケーションへの準拠を可能とするために、そのフォーマットを規定する仕組みを示している。
MIME-type	データ対象物を解釈、処理または表現する能力のあるアプリケーションへの準拠を可能とするために、そのフォーマットを識別する特定の文字列。 IETF は標準 MIME-types の登録を行っている。追加の MIME-types のユーザー定義も可能。
MimeType	ContentItem の MIME-type の指示。
NameLabel	NewsItem の識別の助けになる名前として、人間の使用者が使う記号列。
naming scheme	既知の意味をもつ名前や符号の一群。
news object	News ML 文書の主な構成の一つ。異なる種類の news object は、 NewsEnvelope 、 NewsItem 、 NewsComponent 、 ContentItem である。
NewsComponent	news objects の入れ物。お互いとの関連で news objects の役割を識別するために、メタデータをそれらに基づけるために使用される。
NewsEnvelope	1つ以上の NewsItem を NewsML 文書として送信することについての情報。
NewsIdentifier	NewsItem のための固有の識別子。 ProviderId 、 DateId 、 NewsItemId 、 RevisionId で構成される A4-part 識別子と、これら4つの下位要素のコンポーネントすべてを結びつける PublicIdentifier 。
NewsItem	ニュースの意味ある項目。これは NewsML 文書内の XML element type 。 NewsItem は簡単なものか複雑なものであり、なにかの媒体によるものか媒体の組み合わせによるものであり得る。そのものが NewsItem であるとわかるのは、ある出来事・事件に関し、特定の時間に、ある視点を表す管理情報を加えることによる。このためには最低限、視点を表すための時間とソース（人または組織）を関連づけるのに十分なメタデータが必要である。
NewsItemId	特定の NewsItem について配信社が決める固有の識別子。 NewsItem の同一性の構成を決定し、これに基づき管理された方法で NewsItemId を割当てるのは配信社の側である。
NewsItemRef	NewsItemRef 要素を置き換えるための、外部の NewsItem へのポインタ。
NewsItemType	NewsItem のタイプ指示
newslines	テキストからなる特殊なニュース・メタデータで、ユーザーにニュース・メタデータに関連する NewsItem に関する情報のキーになる項目を提供することを意図している。 NewsLine 自体により運ばれる情報は、 NewsLine 自体により運ばれる情報の一部か、他のニュース・メタデータのいくつかを複写することができる。 NewsLine の例は HeadLine と ByLine 。
NewsLine	NewsML の仕様には含まれないタイプの newslines 。

NewsLines	NewsComponent にあるすべての NewsLine の入れ物。
NewsLineText	ユーザー定義タイプの newsline テキスト。一つの NewsLine に複数のNewsLineText要素があることがあり、それらは言語によって区別される。
NewsLineType	ユーザー定義の NewsLine タイプの表示。
NewsManagement	NewsItem . の管理に関連した情報。
NewsML	NewsML 文書のルート要素。1 つの NewsML 文書は、1 つの NewsEnvelope 及び1つかそれ以上のNewsItemsを包含しなければならない。また、1 つの Catalog 要素及び1つの TopicSet 要素を含むことができる。
NewsProduct	NewsML 文書内ですべての NewsItem が所属するプロダクト（製品）に対する識別子。
NewsService	NewsML 文書内ですべての NewsItem が所属するサービスに対する識別子。
notation（ノーテーション）	一個のデータと、それを翻訳、処理、表現することが可能な一個のアプリケーションとの間の特定の連携。これは XML 仕様書で定義された正式構文。
Notation	ContentItem の記法（ notation ）表示。
OfInterestTo	NewsItem がターゲットとする聴衆の表示。
Origin	テキストのすべてあるいは一部を包むもので、自然言語でここに記述されているものに形式的に一致するデータの項目に対するポインタを含む。
Party	ニュースのワークフローで、この NewsItem に対し特別な関係がある個人、会社、組織の表示。
pointer（ポインタ）	その用途がデータオブジェクトを識別するためのキャラクター文字列。そのオブジェクトにリンクを張るためにか、文書を送信するたびごとにそのオブジェクトそのものを送らず文書内に含めるために用いる。
PreviousRevision	現在の NewsItem の前の版の RevisionId の値。PreviousRevision属性の値は、 NewsItem の前の版があれば、 RevisionId 要素の内容と等しくなければならない。 NewsItem の前の版がなければ0になる。
Priority	NewsItem . の優先表示の指示。
Property	NewsComponent または Topic のプロパティ。プロパティには名前に加え、簡単なValue もしくは後続のプロパティのセットからなる複雑なValueいずれかがある。Value属性がプロパティの値のストリング表現を提供する一方、ValueRef属性は Topic 内にあるかもしくはデータの他の部分にある値を指定する。AllowedValues属性が存在すれば、それはプロパティに与えられた値の限界を定める controlled vocabulary を指定する。
Provider	news object を配信する個人、企業、または組織。
ProviderId	NewsItem . を制作したニュース配信社の個別識別子。 NewsIdentifier のDateId下位要素から識別される日に配信社が保持しているドメイン名か、 controlled vocabulary から引き出される配信社名でなければならない。
public identifier	controlled vocabulary に記述されているか、もしくは、controlled syntaxを使用するリソースを表す文字列。
PublicIdentifier	（XML1.0 仕様書で定義された意味の） NewsItem のための公開識別子。

Rank	NewsComponent 中の BasisForChoice 要素間の優先順位を示す整数。Rank の値が小さい BasisForChoice 要素が大きい値のものより優先される。
raw data	NewsML で定義された構造になってないデータのこと。NewsML アプリケーションを経由してほかのアプリケーションへ、または解読したり、処理することができるユーザーへ転送されなければならない。
Relevance	特定の受信者への NewsItem の関連性の表示。
Repeat	TransmissionId の属性の一つで、前の送信の繰り返しを区別する。
Replace	NewsItem 内の指定された要素を置き換える命令。
Resource	リソースが提供された場所を示し、NewsML 文書で現在のサブツリー内にある formal name の default vocabulary として使用されているかどうかを示す。
RevisionHistory	NewsItem の改訂履歴が入っているファイルへのポインター。
RevisionId	該当する NewsItem の版数を正整数で示す。同じ ProviderId 、 DateId 、 NewsItemId を持つ 2 つのデータ・オブジェクトが同じコンテンツであることを保証するのは配信社の責任である。わずかであっても、 NewsItem を変更して再発行するときは、新しい版には必ず大きい整数の RevisionId を割り当てられなければならない。
RevisionStatus	現在の版にいたる前の版についてのステータスを示す。オプションの Revision 属性は整数であり、該当の版の RevisionId と同じでなければならない。もし、版数がないのであれば、前のすべての版数をステータスに当てはめる。
RightsHolder	使用権 (usage right) を誰が持っているのかを示す文字列。オプションとして、関係する人、会社、組織についてのさらなる情報を、ポインタにより追加することができる。
RightsLine	権利 (right) 情報についての記述であり、著作権 (copyright) 情報とは異なる。著作権情報はニュースオブジェクト所有者について記述するが、権利情報は使用許可を与えた者、その使用方法、使用環境について示す。
RightsMetadata	NewsComponent に関係する権利についてのメタデータ。
Role	NewsComponent 内で、中の NewsComponent が果たす役割の識別子。
schema	XML 文書の階層構造の公式定義。schema はそれ自身が XML 文書であり、 W3C の XML Schema specification と一致する。 DTD で表現するより多くの制約や構造上の法則を持たせることができる。
Scheme	Scheme 属性は、 controlled vocabulary 中に数多くありうる naming schemes のうち、どれが正当に FormalName を管理するものなのかを区別させることに使われる。
SentFrom	NewsML 文書を送信する個人、企業、または組織。
SentTo	NewsML 文書を受信する個人、企業、または組織。
SeriesLine	続き物における news object の位置付けに関する情報について表示用の記述を示す。
SizeInBytes	ContentItem のインラインデータ、または外部参照データの正確なバイト数。

SlugLine	NewsComponent の slug line を表示するのに使われる文字列。ハイパーリンクが張られていたり、書式設定が施されていることもありうる。（"slug line" という用語の意味と使用法については、個々の配信社が、おのおののワークフローと商慣行の範囲内で定義する問題である）
Source	ニュース・オブジェクトの素材を供給した個人、企業、組織もしくはその組み合わせ。
StartDate	指定された使用権の効力が発生する日付を、コンピュータ言語でない普通の言語で記述したもの。
Status	NewsItem のステータス（状態）を示すもの。
StatusWillChange	指定された日時に自動実行されるステータス（状態）の変更の通知をあらかじめ記述する。
subelement	他の要素に含まれる要素。.
	例: <pre><MyElement><Child/><Child/></MyElement></pre> ここでは 2 つの Child という要素が、MyElement という要素の下位要素になる。
SubHeadLine	表示用の補足見出し。
SubjectCode	SubjectCode要素は IPTCSubjectCode のコンテナであり、 IPTC 情報交換モデル（Information Interchange Model=IIM）で定義される NewsItem の内容を示す。それは 1 つまたは複数の Subject、SubjectMatter、SubjectDetail、SubjectQualifier 下位要素で構成される。オプションで 1 つまたは複数の SubjectQualifier要素により拡大される。
Subject	NewsItem の Subject を示す。
SubjectMatter	NewsItem の SubjectMatter を示す。
SubjectDetail	NewsItem の SubjectDetail を示す。
SubjectQualifier	NewsItem の SubjectQualifier を示す。
system identifier	システム上で、リソースの場所を特定するアドレス。これは一般的には、絶対的あるいは相対的なファイルのパスもしくは URI 。
SystemIdentifier	NewsItem の（XML1.0 仕様書で定義された意味の）システム識別子。
ThisRevisionCreated	NewsItem の最新版が作成された日付。オプションで時間が入る。ISO8601 の基本フォーマットで表現される。
topic	ニュースで言及されている実世界のあらゆる事象もしくは概念。topicの例としてはイラン・イラク戦争、トニー・ブレア、パキスタン首相、IBM、国連、ダイソン電気掃除機、中国、クルディスタン、パリ、クレムリン、エイズ、アスピリンなど
topic reference	Directory内で topic へのポインタの役割を果たす要素。
Topic	NewsComponent において、正式に命名された事象(topic)もしくは出来事に関する情報を提供する element 。Topicは 1 つまたはそれ以上のTopicType

	subelementsをもたなければならない。そのTopicType subelementsはTopicのタイプを表す。
TopicOccurrence	NewsComponent のコンテンツで、あるTopicが発生することの表示。
TopicSet	Topics の収納場所。
TopicSetRef	最新のものと併合させられるべき TopicSet のポインタ。
TopicType	Topic のタイプの表示。
TopicUse	特定の Topicが NewsML ドキュメントのどこに使われているかを示すもの。
TransmissionId	NewsML ドキュメント送信のための唯一の識別子。
Update	既存の NewsItem の修正。挿入、置換や削除。
Urgency	NewsItem の緊急性の識別子。
Url	Resource の位置特定に使える URL。
Urn	リソースに世界レベルの識別子を与える URN。 PublicIdentifier で記述したように、これは一般的に（必ずしもというわけではないが）NewsML の URN となるだろう。
UsageRights	NewsComponent に属する使用权に関する情報を付与する。その UsageType 、 Geography 、 RightsHolder 、 Limitations 、 StartDate 、 EndDate という下位要素は、普通の言語で書かれたメタデータを追加する。
UsageType	自然言語により権利が適用される利用のタイプを示す。
URI	Uniform Resource Indicator の略。特定のリソースを識別する（時には、場所を特定する）のに使われる、世界で一つしかない文字列。URL もしくは URN であろう。
URL	Uniform Resource Locator の略。本来はウェブ上でリソースを見つけることのできるアドレスである。また、これはウェブ・リソースの識別子で、http://のプロトコルがウェブ・リソースを識別し、それにアクセスするのに使われる。
URN	Uniform Resource Name。現在の場所とは無関係に、リソースの場所を特定する世界的に固有の文字列。
UTC	Coordinated Universal Time。協定世界時。国際タイムビューローが定義した時間の基準で、標準的な周波数と時間の信号発信の基礎となる。UTC は（不正確に）グリニッチ標準時（GMT）と呼ばれることも多い。
Value	Property タグの属性。 Property の値の文字列表現。
ValueRef	Property の値へのポインタ。 TopicSet 中の Topic もしくはデータの他の部分。
Variant	Description タグのオプションの属性。タグの属性。同一言語で異なった表現をし、それらを区別する場合に使用する。.
Version	NewsML の任意な属性で、文章を識別する DTD や Schema のバージョンを与える。
Vocabulary	ボキャブラリー属性は、 FormalName の意味を解くために使える controlled vocabulary であるカレント文書中の TopicSet を識別する。

W3C	World Wide Web Consortium
XML	Extensible Markup Language 。W3Cが1998年2月に勧告。
xml:lang	特別な属性 (attribute) で、XML specification (仕様書) で定義されている。 XML 要素の内容の言語を特定するために使用する。その値はISOで定める言語コードに従う。
XPath	XML Path Language 。W3Cが1999年11月に勧告。特定のXML文書中のオブジェクトを参照する方法を規定する。
XPointer	XML Pointer Language 。W3Cの2000年6月勧告案(標準化作業中)。汎用的なXML文書中のオブジェクトを参照する方法を規定する。
XSLT	XML Stylesheet Language (Transformations) 。W3C が1999年11月に勧告。

7 付録

7.1 日本新聞協会 NewsML レベル 1.2 ガイドライン表

◎ 必須／○ 省略可／△ 規定外／× 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
NewsML	-					
		Version	CDATA	#IMPLIED	◎	*
Catalog?	-				◎	
		Href	CDATA	#IMPLIED	○	
Resource*	-				○	
Um?	#PCDATA				○	
Url*	#PCDATA			1	○	
DefaultVocabularyFor*	EMPTY				○	
		Context	CDATA	#REQUIRED	○	
		Scheme	CDATA	#IMPLIED	△	
TopicUse*	EMPTY				△	
		Topic	CDATA	#REQUIRED	△	
		Context	CDATA	#IMPLIED	△	
TopicSet*	-				△	
		FormalName	CDATA	#REQUIRED	△	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
Comment*	#PCDATA				△	
		xml:lang	CDATA	#IMPLIED	△	
		TranslationOf	IDREF	#IMPLIED	△	
		FormalName	CDATA	#IMPLIED	△	*
		Vocabulary	CDATA	#IMPLIED	△	*
		Scheme	CDATA	#IMPLIED	△	*
Catalog?	↑				△	
TopicSetRef*	-				△	
		TopicSet	CDATA	#IMPLIED	△	
Comment*	↑				△	
Topic*	-				△	
		Details	CDATA	#IMPLIED	△	
Comment*	↑				△	
Catalog?	↑				△	
TopicType+	EMPTY				△	
		FormalName	CDATA	#REQUIRED	△	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
FormalName*	#PCDATA				△	
		Scheme	CDATA	#IMPLIED	△	
Description*	#PCDATA				△	
		xml:lang	CDATA	#IMPLIED	△	
		Variant	CDATA	#IMPLIED	△	
Property*	-				X	
		FormalName	CDATA	#REQUIRED	X	
		Vocabulary	CDATA	#IMPLIED	X	
		Scheme	CDATA	#IMPLIED	X	
		AssignedBy	CDATA	#IMPLIED	X	
		Importance	CDATA	#IMPLIED	X	
		Confidence	CDATA	#IMPLIED	X	
		HowPresent	CDATA	#IMPLIED	X	
		DateAndTime	CDATA	#IMPLIED	X	
		Value	CDATA	#IMPLIED	X	
		ValueRef	CDATA	#IMPLIED	X	
		AllowedValues	CDATA	#IMPLIED	X	
		AllowedScheme	CDATA	#IMPLIED	X	*
Property*	↑				X	

◎ 必須／○ 省略可／△ 規定外／× 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
NewsEnvelope	-				◎	
TransmissionId?	#PCDATA				○	
Repeat		Repeat	CDATA	#IMPLIED		
SentFrom?	-				○	
Comment*	↑				△	
Party+	-				○	
FormalName		FormalName	CDATA	#REQUIRED	○	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
Topic		Topic	CDATA	#IMPLIED	△	
Property*	↑				×	*
SentTo?	-				○	
Comment*	↑				△	
Party+	↑				○	
DateAndTime	#PCDATA				◎	
NewsService*	EMPTY				○	
FormalName		FormalName	CDATA	#REQUIRED	○	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
NewsProduct*	EMPTY				◎	
FormalName		FormalName	CDATA	#REQUIRED	◎	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
Priority?	EMPTY				○	
FormalName		FormalName	CDATA	#REQUIRED		
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED		
NewsItem+	-				◎	
xml:lang		xml:lang	CDATA	#IMPLIED	○	
※		※			○	
TranslationOf		TranslationOf				
属性値は“△”						
Comment*	↑					
Catalog?	↑				○	
Identification					◎	
NewsIdentifier	-				◎	
ProviderId	#PCDATA				◎	
Vocabulary		Vocabulary	CDATA	#IMPLIED	X	
DateId	#PCDATA				◎	
NewsItemId	#PCDATA				◎	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
RevisionId	#PCDATA				◎	
PreviousRevision		PreviousRevision	CDATA	#REQUIRED	◎	
Update		Update	CDATA	#REQUIRED	◎	
PublicIdentifier	#PCDATA				◎	
NameLabel?	#PCDATA				X	
DateLabel?	#PCDATA				X	
Label*	-				X	
LabelType	EMPTY				X	
FormalName		FormalName	CDATA	#REQUIRED	X	
Vocabulary		Vocabulary	CDATA	#IMPLIED	X	
Scheme		Scheme	CDATA	#IMPLIED	X	
LabelText	#PCDATA				X	

◎ : 必須 / ○ : 省略可 / △ : 規定外 / × : 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
NewsManagement	-				◎	
NewsItem	EMPTY				◎	
FormalName		FormalName	CDATA	#REQUIRED	◎	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
FirstCreated	#PCDATA				◎	
ThisRevisionCreated	#PCDATA				◎	
Status	EMPTY				◎	
FormalName		FormalName	CDATA	#REQUIRED		
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
StatusWillChange*	-				△	
FutureStatus	EMPTY				△	
FormalName		FormalName	CDATA	#REQUIRED	△	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
DateAndTime	#PCDATA				△	
Urgency?	EMPTY				△	
FormalName		FormalName	CDATA	#REQUIRED	△	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
RevisionHistory?	EMPTY				X	
Href		Href	CDATA	#REQUIRED	X	
DerivedFrom*	-				○	
FormalName		FormalName	CDATA	#IMPLIED	×	*
Vocabulary		Vocabulary	CDATA	#IMPLIED	×	*
Scheme		Scheme	CDATA	#IMPLIED	×	*
NewsItem		NewsItem	CDATA	#IMPLIED	○	
Comment*	↑				○	
AssociatedWith*	-				○	
FormalName		FormalName	CDATA	#IMPLIED	×	*
Vocabulary		Vocabulary	CDATA	#IMPLIED	×	*
Scheme		Scheme	CDATA	#IMPLIED	×	*
NewsItem		NewsItem	CDATA	#IMPLIED	○	
Comment*	↑				○	
Instruction*	-				△	
FormalName		FormalName	CDATA	#REQUIRED		
Vocabulary		Vocabulary	CDATA	#IMPLIED		
Scheme		Scheme	CDATA	#IMPLIED		
RevisionStatus*	-					
Revision		Revision	CDATA	#IMPLIED		
Status	↑					
Property*	↑				X	

◎ 必須／○ 省略可／△ 規定外／× 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
NewsComponent	-	Essential	(yes/no)	no	○	
		EquivalentsList	(yes/no)	no	△	
		xml:lang	CDATA	#IMPLIED	○	
Comment*	↑				○	
Catalog?	↑				○	
TopicSet*	↑				△	
Role?	EMPTY				△	
		FormalName	CDATA	#REQUIRED		
		Vocabulary	CDATA	#IMPLIED		
		Scheme	CDATA	#IMPLIED		
BasisForChoice*	#PCDATA				△	
		Rank	CDATA	#IMPLIED		
NewsLines?	-				○	
HeadLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED	○	
Origin	-				△	
		AssignedBy	CDATA	#IMPLIED		
		Importance	CDATA	#IMPLIED		
*		Confidence	CDATA	#IMPLIED		
		HowPresent	CDATA	#IMPLIED		
		DateAndTime	CDATA	#IMPLIED		
		Href	CDATA	#IMPLIED		
SubHeadLine*	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
ByLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
* ByLineTitle*	(#PCDATA Origin)*				×	*
		xml:lang	CDATA	#IMPLIED		
DateLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
CreditLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
CopyrightLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
RightsLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
SeriesLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
SlugLine	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED		
KeywordLine	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		
NewsLine	-				○	
NewsLineType	EMPTY				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED		
		Scheme	CDATA	#IMPLIED		
NewsLineText+	(#PCDATA Origin)*				○	
		xml:lang	CDATA	#IMPLIED		

◎ 必須／○ 省略可／△ 規定外／× 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
AdministrativeMetadata?	-				○	
Catalog?	↑				○	
FileName?	#PCDATA				△	
SystemIdentifier?	#PCDATA				△	
Provider?	-				○	
Comment*	↑				△	
Party+	↑	※SentFromと同じ			○	
Creator*	-				○	
Comment*	↑				△	
Party+	↑	※SentFromと同じ			○	
Contribution*	EMPTY				×	*
		FormalName	CDATA	#REQUIRED	×	*
		Vocabulary	CDATA	#IMPLIED	×	*
		Scheme	CDATA	#IMPLIED	×	*
Source*	-				△	
		NewsItem	CDATA	#IMPLIED	△	
Comment*	↑				△	
Party+	↑	※SentFromと同じ			△	
Contributor*	-				△	
Comment*	↑				△	
Party+	↑	※SentFromと同じ			△	
Contribution*	EMPTY				×	*
		FormalName	CDATA	#REQUIRED	×	*
		Vocabulary	CDATA	#IMPLIED	×	*
		Scheme	CDATA	#IMPLIED	×	*
Property*	↑				×	
RightsMetadata?	-				○	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED		
		Confidence	CDATA	#IMPLIED		
		HowPresent	CDATA	#IMPLIED		
		DateAndTime	CDATA	#IMPLIED		
Catalog?	↑				○	
Copyright*	-				○	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED		
		Confidence	CDATA	#IMPLIED		
		HowPresent	CDATA	#IMPLIED		
		DateAndTime	CDATA	#IMPLIED		
Comment*	↑				△	
CopyrightHolder	(#PCDATA Origin)*				○	
	xml:lang	CDATA	#IMPLIED		○	
CopyrightDate	(#PCDATA Origin)*				○	
	xml:lang	CDATA	#IMPLIED		○	

◎ 必須／○ 省略可／△ 規定外／× 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
UsageRights*	-				△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
UsageType?	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
Geography?	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
RightsHolder?	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
Limitations?	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
StartDate?	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
EndDate?	(#PCDATA Origin)*				△	
		xml:lang	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
Property*	↑				X	

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
DescriptiveMetadata?	-				○	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED		
		Confidence	CDATA	#IMPLIED		
		HowPresent	CDATA	#IMPLIED		
		DateAndTime	CDATA	#IMPLIED		
	Catalog?				○	
	Language*				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
?		Scheme	CDATA	#IMPLIED		
		AssignedBy	CDATA	#IMPLIED		
		Importance	CDATA	#IMPLIED		
		Confidence	CDATA	#IMPLIED		
		HowPresent	CDATA	#IMPLIED		
		DateAndTime	CDATA	#IMPLIED		
	Genre*				X	
		FormalName	CDATA	#REQUIRED	X	
		Vocabulary	CDATA	#IMPLIED	X	
		Scheme	CDATA	#IMPLIED		
		AssignedBy	CDATA	#IMPLIED		
		Importance	CDATA	#IMPLIED		
		Confidence	CDATA	#IMPLIED		
		HowPresent	CDATA	#IMPLIED		
		DateAndTime	CDATA	#IMPLIED		
	SubjectCode*				○	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
	Subject				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
	SubjectMatter				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
	* SubjectDetail				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	
	SubjectQualifier?				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	△	
		Importance	CDATA	#IMPLIED	△	
		Confidence	CDATA	#IMPLIED	△	
		HowPresent	CDATA	#IMPLIED	△	
		DateAndTime	CDATA	#IMPLIED	△	

◎ 必須／○ 省略可／△ 規定外／× 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
OfInterestTo*	-				X	
		FormalName	CDATA	#REQUIRED	X	
		Vocabulary	CDATA	#IMPLIED	X	
		Scheme	CDATA	#IMPLIED	X	
		AssignedBy	CDATA	#IMPLIED	X	
		Importance	CDATA	#IMPLIED	X	
		Confidence	CDATA	#IMPLIED	X	
		HowPresent	CDATA	#IMPLIED	X	
		DateAndTime	CDATA	#IMPLIED	X	
Relevance?	EMPTY				X	
		FormalName	CDATA	#REQUIRED	X	
		Vocabulary	CDATA	#IMPLIED	X	
		Scheme	CDATA	#IMPLIED	X	
		AssignedBy	CDATA	#IMPLIED	X	
		Importance	CDATA	#IMPLIED	X	
		Confidence	CDATA	#IMPLIED	X	
		HowPresent	CDATA	#IMPLIED	X	
		DateAndTime	CDATA	#IMPLIED	X	
DateLineDate?	#PCDATA				×	*
Location*	-				×	*
		AssignedBy	CDATA	#IMPLIED	×	*
		Importance	CDATA	#IMPLIED	×	*
		Confidence	CDATA	#IMPLIED	×	*
		HowPresent	CDATA	#IMPLIED	×	*
		DateAndTime	CDATA	#IMPLIED	×	*
		xml:lang	CDATA	#IMPLIED	×	*
		Topic	CDATA	#IMPLIED	×	*
Property*	↑				×	*
TopicOccurrence*	EMPTY				X	
		AssignedBy	CDATA	#IMPLIED	X	
		Importance	CDATA	#IMPLIED	X	
		Confidence	CDATA	#IMPLIED	X	
		HowPresent	CDATA	#IMPLIED	X	
		DateAndTime	CDATA	#IMPLIED	X	
		Topic	CDATA	#IMPLIED	X	
Property*	↑				X	
Metadata*					○	
Catalog?	↑				○	
MetadataType	EMPTY				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
Property+	↑				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
		AssignedBy	CDATA	#IMPLIED	X	
		Importance	CDATA	#IMPLIED	X	
		Confidence	CDATA	#IMPLIED	X	
		HowPresent	CDATA	#IMPLIED	X	
		DateAndTime	CDATA	#IMPLIED	X	
		Value	CDATA	#IMPLIED	○	
		ValueRef	CDATA	#IMPLIED	×	
		AllowedValues	CDATA	#IMPLIED	△	
		AllowedScheme	CDATA	#IMPLIED	△	*

◎ : 必須 / ○ : 省略可 / △ : 規定外 / × : 使用停止

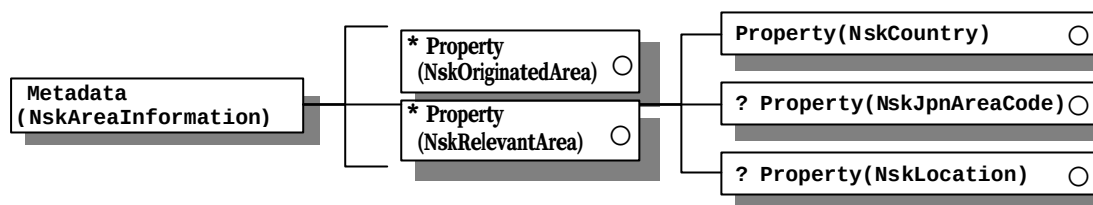
タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
NewsItem+	↑				X	
NewsItemRef+	-				X	
NewsItem		NewsItem	CDATA	#IMPLIED	X	
Comment*	↑				X	
NewsComponent+	↑				X	
ContentItem+	-				○	
Href		Href	CDATA	#IMPLIED	○	
Comment*	↑				△	
Catalog?	↑				○	
MediaType?	EMPTY				○	
FormalName		FormalName	CDATA	#REQUIRED	○	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
Format?	EMPTY				○	
FormalName		FormalName	CDATA	#REQUIRED	○	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
MimeType?	EMPTY				○	
FormalName		FormalName	CDATA	#REQUIRED	○	
Vocabulary		Vocabulary	CDATA	#IMPLIED	△	
Scheme		Scheme	CDATA	#IMPLIED	△	
Notation?	EMPTY				X	
FormalName		FormalName	CDATA	#REQUIRED	X	
Vocabulary		Vocabulary	CDATA	#IMPLIED	X	
Scheme		Scheme	CDATA	#IMPLIED	X	
Characteristics?	-				○	
SizeInBytes?	#PCDATA				○	
Property*	↑				△	
Encoding	-				△	
Notation		Notation	CDATA	#REQUIRED	△	
Encoding	↑				△	
DataContent	ANY				△	
DataContent	ANY				○	
Update+					×	
InsertBefore	ANY				×	
DuidRef		DuidRef	CDATA	#REQUIRED		
InsertAfter	ANY				×	
DuidRef		DuidRef	CDATA	#REQUIRED		
Replace	ANY				×	
DuidRef		DuidRef	CDATA	#REQUIRED		
Delete	EMPTY				×	
DuidRef		DuidRef	CDATA	#REQUIRED		
TopicSet	↑				○	

◎ : 必須 / ○ : 省略可 / △ : 規定外 / × : 使用停止

タグ名	値	属性名	属性値	省略値	1.2	1.2の変更点
TopicSet*	-				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
Comment*	#PCDATA				△	
*		xml:lang	CDATA	#IMPLIED	△	
		TranslationOf	IDREF	#IMPLIED	△	
		FormalName	CDATA	#IMPLIED	△	*
		Vocabulary	CDATA	#IMPLIED	△	*
		Scheme	CDATA	#IMPLIED	△	*
Catalog?	↑				△	
TopicSetRef*	-				△	
		TopicSet	CDATA	#IMPLIED	△	
Comment*	↑				△	
Topic*	-				○	
		Details	CDATA	#IMPLIED	△	
Comment*	↑				△	
Catalog?	↑				△	
TopicType+	EMPTY				○	
		FormalName	CDATA	#REQUIRED	○	
		Vocabulary	CDATA	#IMPLIED	△	
		Scheme	CDATA	#IMPLIED	△	
FormalName*	#PCDATA				○	
		Scheme	CDATA	#IMPLIED	○	
Description*	#PCDATA				○	
		xml:lang	CDATA	#IMPLIED	○	
		Variant	CDATA	#IMPLIED	○	
Property*	-				X	
		FormalName	CDATA	#REQUIRED	X	
		Vocabulary	CDATA	#IMPLIED	X	
		Scheme	CDATA	#IMPLIED	X	
		AssignedBy	CDATA	#IMPLIED	X	
		Importance	CDATA	#IMPLIED	X	
		Confidence	CDATA	#IMPLIED	X	
		HowPresent	CDATA	#IMPLIED	X	
		DateAndTime	CDATA	#IMPLIED	X	
		Value	CDATA	#IMPLIED	X	
		ValueRef	CDATA	#IMPLIED	X	
		AllowedValues	CDATA	#IMPLIED	X	
		AllowedScheme	CDATA	#IMPLIED	×	*
Property*	↑				X	

7.2 地域情報の表現方法について

[ツリー図]



[NSK 解説]

NewsML には、利用権に関する地域情報を記述するための UsageRights/Geography 要素があるが、新聞協会が必要と考える、ニュースの発生場所やニュースの関連地域という意味での地域表現が可能な要素はない。Metadata 要素の Property 下位要素を用いて記述する方法を新聞協会として規定する。そのため、新聞協会で地域情報用の MetadataType を定義し、Metadata 要素の Property 下位要素を用いて記述することにした。この MetadataType 要素の FormalName 属性値は NskAreaInformation である。

[NskNewsML:1 記述形式]

MetadataType の FormalName 属性値が NskAreaInformation である Metadata 要素内の Property 要素を使用して地域・場所情報を記述する。

Property 要素を 2 階層（親子関係）で表現する。

親要素は、ニュースの発生場所、ニュースの関連する地域、という意味、用途の種別を記述する。下位要素には具体的な地域の値を記述する。

地域情報を表す Property のボキャブラリは NSK を使用する。以下の FormalName がある。

NskOriginatedArea	親要素に使用でき、ニュースの発生場所を表現する。 この FormalName を持つ親要素は繰り返しなし。
NskRelevantArea	親要素に使用でき、ニュースの関連する地域を表現する。 この FormalName を持つ親要素は繰り返しあり。
NskCountry	国を表現する。Value 属性には、IPTC のボキャブラリに記述されている値を使用する。 この FormalName を持つ下位要素は省略不可で繰り返しなし。
NskJpnAreaCode	都道府県／市区町村コードを表現する。NSK オリジナルボキャブラリに記述されている値を使用する。 この FormalName を持つ下位要素は省略可で繰り返しなし。
NskLocation	自由な表現で場所を説明する。 この FormalName を持つ下位要素は省略可で繰り返しなし。

なお、1 つのニュースに関連する地域が複数ある場合については、親の Property を並列に表記して表現する。

(例)

「ニュースの発生場所」と「ニュースの関連地域」について表現する方法を例示する。

○ニュースの発生場所についてのサンプル 1

下記は、ニュースの発生場所が「アメリカ合衆国、ハワイ島ホノルル沖」であることを表している。

```
<Property FormalName="NskOriginatedArea">
  <Property FormalName="NskCountry" Value="USA"/>
  <Property FormalName="NskLocation" Value="ハワイ島ホノルル沖"/>
</Property>
```

○ニュースの発生場所についてのサンプル 2(日本国内の表現)

下記は、ニュースの発生場所が「日本、愛媛県宇和島市、宇和島水産高校」であることを表している。

```
<Property FormalName="NskOriginatedArea">
  <Property FormalName="NskCountry" Value="JPN"/>
  <Property FormalName="NskJpnAreaCode" Value="38203"/>
  <Property FormalName="NskLocation" Value="宇和島水産高校"/>
</Property>
```

上記の例は、「NskOriginatedArea(ニュースの発生場所)」が“日本”(NskCountry：国を表すプロパティ)であり、“38203”(NskJpnAreaCode：都道府県コードを表すプロパティ。38203は愛媛県宇和島市)であり、“宇和島水産高校”(NskLocation：自由形式)であることを表している。子 Property 要素は並列表記されていることからわかるとおり、“日本”と“38203”は特に関係を持っていない。“日本の中の38203(愛媛県宇和島市)”という包含関係は表記されていないので、発生場所は「日本であり」「38203である」ことを表現しているだけであることに注意すべきである。

○複数の関連地域がある場合

下記は、ニュースの関連地域が、「アメリカ合衆国」と「日本、愛媛県宇和島市」であることを表している。

```
<Property FormalName="NskRelevantArea">
  <Property FormalName="NskCountry" Value="USA"/>
</Property>

<Property FormalName="NskRelevantArea">
  <Property FormalName="NskCountry" Value="JPN"/>
  <Property FormalName="NskJpnAreaCode" Value="38203"/>
</Property>
```


7.3 NewsML 複合ガイドライン

[NSK 解説]

NewsML では、`NewsItem`、`NewsComponent`、`ContentItem` それぞれを複数記述してニュースや素材を複合することができる。

複合の方法は次のタイプにわかれる

NewsML 内での `NewsItem` 列挙

複数の `NewsItem` (ニュースやコンテンツ) を 1 つの NewsML ファイルに統合する場合に用いる。ただし、それぞれの `NewsItem` 同士の関連性などは表現ができない。

具体的には、記事の `NewsItem` と写真の `NewsItem` を 1 つの NewsML に入れただけでは、関連素材とは見なされない。後述の他の方法を用いるか、関連性をもたせるためには `AssociatedWith` などの機能を使う必要がある。`AssociatedWith` で関連を示すのであれば無理に `NewsItem` を 1 つの NewsML に入れる必要は無い。

NewsComponent 内での `NewsComponent` 列挙

複数の関連素材を 1 つにまとめる目的で利用される。次の `ContentItem` の方法に比べ、`NewsComponent` が持つ多くのメタ情報を利用できる。

ただし、NewsML でのコンテンツの管理単位は `NewsItem` であり、`NewsComponent` 単位での管理ができない。このため、常に複数をまとめて管理することが前提となってしまう。

考えられる組み合わせとしては、写真とキャプション、など常にセットで管理するコンテンツや、変更の必要の無い長期的なデータベースでの記事/写真の組み合わせが考えられる。また、html の記事とプレーンテキストの記事、写実画像とサムネイル、等が考えられるが、次の `ContentItem` 列挙でも実現できる。

NewsComponent 内での `ContentItem` 列挙

1 つの素材で複数データを持ったり、1 つのものを複数種類で表現するデータがある場合などに用いる。

具体的には、1 つの画像に関する「実画像」と「サムネイル」、1 つの記事の「プレーンテキスト」と「html」、また、「音声」と「テキスト」、「映像」と「サムネイル」などの組み合わせがある。「記事」と「関連写真」などの組み合わせは `ContentItem` のメタ情報では表現が難しいため、上述したの `NewsComponent` 列挙や、下記の `NewsItem` の列挙などが必要になる。

NewsComponent 内での `NewsItem` (`NewsItemRef`) 列挙

複数の `NewsItem` を 1 つに関連付ける方法として用意されている。記事の `NewsItem` と画像の `NewsItem` などを組み合わせるものとして利用できる。

これは編成 (ニュースコンテンツを組み合わせる) という場面が分かりやすい。それぞれ別で管理されている `NewsItem` の素材を、1 つの NewsML ニュースに統合するといった処理になる。上述した `NewsComponent` 列挙でも表現できるが、組み合わせた後にそれぞれのコンテンツを編集して、バージョンアップしたり、場合によっては同じコンテンツを共有する場合は `NewsItem` に分かれていたほうが都合が良い。最新版をポイントするにはリビジョンを省略すれば常に最新の組み合わせが表現できるなどです。

配信フォーマットとしてというよりは、コンテンツの編集や編成のための素材データ管理の仕組として使い勝手がよい。新聞のシステムで言えば、大組面を管理する NewsML に NewsItemRef として、その面で利用する素材の NewsML へのリンク複数持たせるような使い方である。

NewsItem と NewsItemRef の両方がある、通常は物理的にファイルが分かれる場合があるため NewsItemRef になり、なんらかの要件で 1 つのファイルにまとめる場合は NewsItem の列挙を用いることになる。

[NskNewsML:1 記述形式]

下記の複合形式を(△)として利用できる。

それ以外の複合方法は(×)。

ContentItem の列挙は、NewsComponent 要素の EquivalentList 属性が"yes"の場合のみ利用可能。

NewsComponent を 2 階層まで利用可能。

下層(子)の NewsComponent の列挙は、上層の NewsComponent 要素の EquivalentList 属性"yes"と"no"の両方で利用可能。

下層(子)の NewsComponent は、NskNewsML:1 の基本形に従う。この NewsComponent にも ContentItem の列挙が可能。

各 NewsComponent の各メタ情報(NewsLines, AdministrativeMetadata, RightsMetadata, DescriptiveMetadata, Metadata) は、他 NewsComponent へは継承されない、各 NewsComponent 毎にそれぞれ適切なメタ情報を記述する。

サンプル (ContentItem の列挙) ～プレーンテキストと XML 記事の複合～

```
<NewsComponent EquivalentList=" yes " >
...
  <AdministrativeMetadata>
    ...この素材のProvider (提供元)、Creator (編集元) など...
    ... 上位NewsItemの情報としても扱われる...
  </AdministrativeMetadata>
  ...
  <RightsMetadata>
    ... この素材の権利情報...
</ RightsMetadata>
  <ContentItem>
    <MediaType FormalName="Text" />
    <MimeType FormalName=" text/plain " />
    <DataContent>
      ...プレーンテキスト記事...
    </DataContent>
  </ContentItem>
  <ContentItem>
    <MediaType FormalName="Text" />
    <Format FormalName=" NskBasicText " />
    <MimeType FormalName=" text/xml " />
    <DataContent>
      <body>
        <p>...XMLによる記事のパラグラフ...</p>
      </body>
    </DataContent>
  </ContentItem>
</NewsComponent>
```

サンプル（NewsComponent の列挙）～記事と写真の複合～

```
<NewsComponent EquivalentsList="no" >
  <!-- NewsItemとこのNewsComponentの各種メタ情報 -->
  ...
  <AdministrativeMetadata>
    ...このニュースのProvider（提供元）、Creator（編集元）など…
    ... 上位NewsItemの情報としても扱われる…
  </AdministrativeMetadata>
  ...
  <RightsMetadata>
    ... このニュースの権利情報…
  </ RightsMetadata>
  <!-- 記事のコンポーネント（NskNewsML:1の単体と同じ形態） -->
  <NewsComponent>
    <!-- 記事に関する各種メタ情報 -->
    ...
    <AdministrativeMetadata>
      ...この記事のProvider（提供元）、Creator（編集元）など…
    </AdministrativeMetadata>
    ...
    <RightsMetadata>
      ... この記事の権利情報…
    </ RightsMetadata>
    <ContentItem>
      <MediaType FormalName="Text" />
      ... この記事の物理メタ情報(タイプやフォーマット)と生データ ...
    </ContentItem>
  </NewsComponent>

  <!-- 写真のコンポーネント（NskNewsML:1の単体形態） -->
  <NewsComponent>
    <!-- 写真に関する各種メタ情報 -->
    ...
    <AdministrativeMetadata>
      ...この写真のProvider（提供元）、Creator（作成元）など…
    </AdministrativeMetadata>
    ...
    <RightsMetadata>
      ... この写真の権利情報…
    </ RightsMetadata>
    <ContentItem Href="photo.img">
      <MediaType FormalName="Photo" />
      ... この写真の物理メタ情報(タイプやフォーマット) ... </ContentItem>
    </ContentItem>
  </NewsComponent>
</NewsComponent>
```

7.4 NewsML データ伝送モデルの例

NewsML を企業間やシステム間で伝達するためのデータ伝送モデルの例を下記のバリエーションで説明する。

この説明によって、特にプロトコルやシステムへの実装を規定するものではない。

NewsML には、1つの NewsML ファイルだけからなるものと、外部ファイルを伴うものがある。

また、外部ファイルを伴う場合はその送り方として下記のようなバリエーションを説明する。

単体ファイル伝送

外部ファイルを伴う複合データ伝送

PUSH&PULL による複合データ伝送

PUSH による複合データ伝送

複合データを 1 つにまとめて PUSH する伝送

下記の説明の中では、PUSH と PULL というデータ伝送方法で説明をしている。

PUSH とは、送り手が能動的にデータを伝送する処理を表す。

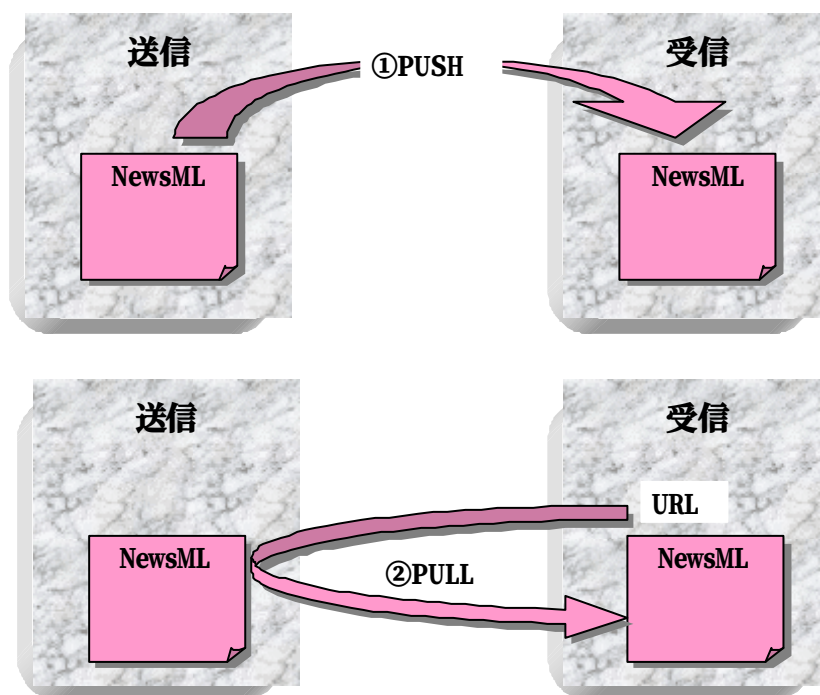
PULL とは、受け手が送り手のシステムからデータを取り出す処理を表す。PULL する場合には、そのデータの所在を知る必要がある。説明の中では、その所在を URL として表している。

URL はプロトコルやインターネットアドレスを含んだもの(<http://アドレス/ファイルパス>)と、ファイルパスだけを記述したものがある。

7.4.1 NewsML 単体ファイル伝送

単体の NewsML ファイル伝送は、1回の PUSH で送ることができる。

別の方法として、受け手は URL をなんらかの手段で入手し、ファイルを PULL する方法もある。



7.4.2 外部ファイルを伴う複合データ伝送

NewsML で写真等のバイナリデータを扱う場合、このバイナリデータは NewsML と別ファイルにすることがある。この場合、外部ファイルを含む、複数のファイルを伝送しなければならない。

ここでは、複数ファイルから構成される複合データの伝送方式として、「PUSH+PULL」、「PUSH」、「1つにまとめてPUSH」の3種類のモデルについて説明する。

説明中で、親と子という表現を使っている。親は NewsML ファイルで子の所在(URL 等)をもっており、親のファイルで子ファイルが特定できることを前提にしている。

7.4.2.1 PUSH&PULL による NewsML 複合データ伝送

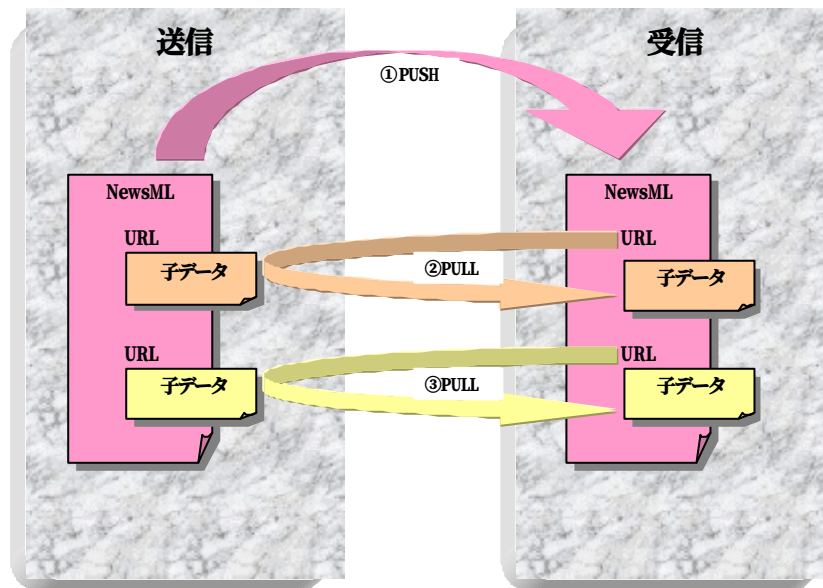
処理概要

送り手は、複合データの親ファイル(NewsML)を PUSH にて伝送。受け手は、親ファイル(NewsML)に記述されている URL を使って子ファイルを PULL で取り出す方式。

送り手が親に記述する URL は、プロトコルとアドレスを含んだ形式で記述しなければならない。また、送り手は、URL に示された場所にデータをあらかじめ準備しておく必要がある。

場合によっては 子ファイルに URL (孫データ)を含む場合も考えられるため、受信側は、再帰的に PULL の処理を行うこともあり得る。

また、他の方法として単体ファイル伝送で書いたように、送り手が親の URL だけを伝送し、受け手はその URL から親ファイルを PULL、さらに親ファイル中の URL を使って、その子ファイルを PULL する方法もある。



利用シーンと特徴

送り手と受け手が常時通信できる環境での利用。

後述する方法にくらべ送り手の処理が単純であり多くのあて先に親ファイルを早く送ることができる。

特殊な配信システムを持たずに Web サーバシステムでデータの供給ができる。

子ファイルの受信失敗時のリトライを受け手側で行うことができる。

受けて側は親ファイルを先に受信するため、受信状況を把握できる。

Web のキャッシュサーバや Web サーバの分散を行って子ファイルの伝送のトラフィックをコントロールしやすい。

受け手が必要に応じて（子ファイルが必要な場合のみ）オンデマンドでデータを受けたい場合に有効。

常時接続ではない接続（ダイアルアップ接続など）の場合、送り手が子ファイルの伝送を制御できないため接続を切るタイミングを受け手から送り手に伝える手順を追加する必要がある。

7.4.2.2 PUSH による NewsML 複合データ伝送

処理概要

送り手は、すべての子ファイルを PUSH し受け手の一時作業フォルダに伝送する。

次に送り手は、親ファイルを受け手が親だと認識できる方法で伝送、受け手は親ファイルの受信を検知して受信処理を開始する方式。

送り手が送るデータ中の URL は、プロトコルやアドレスを含まないファイルパスだけを記述することになる。

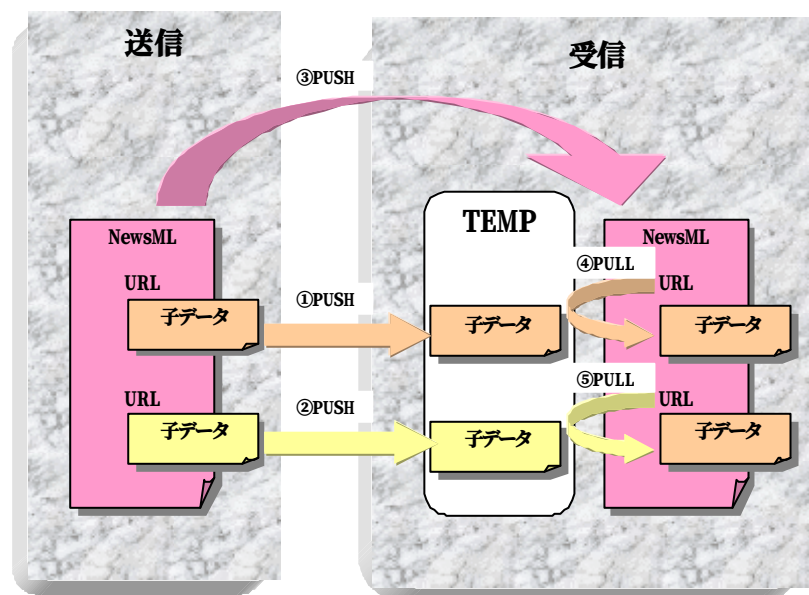
ここで、受け手が親ファイルであることを認識するためには、下記のような方式が考えられる。

親ファイルだと判別できるように、ファイル名に規則を持たせる。

親ファイルを、特別なフォルダ（監視フォルダ・ホットフォルダとも呼ばれる）に送る。

親ファイルのファイル名を最後に伝えるマークファイル等を送る

親ファイルの URL を最後に PUSH する（この場合親子の送る順番は自由）



利用シーンと特徴

送り手と受け手は、常時接続でなくてもかまわないため、ダイアルアップ環境でも利用できる。

親ファイル(NewsML)が最後に送られてくるため、伝送中のデータがどんなデータか判断できない。

送り手が多数の相手に伝送する場合、最後に送る親ファイルは比較的小さいため、子ファイルの伝送にズレがあった場合でも、ほぼ同時に伝送を完了することができ、仮想的な同報ができる。

送り手は、あらかじめ親子関係などを再帰的に分析して必要な子ファイルを送る必要がある。

送り手が伝送失敗を検出できない場合、受信側では再受信する方法が無い、別の手順を使って再伝送を行うか、伝送保証をする仕組みが必要。

7.4.2.3 複合データを1つにまとめてPUSHする伝送

処理概要

送り手は、複合データのすべてのデータを1度にPUSHする方法。

受け手は、受信したデータを、元の複数ファイルに戻して利用する。

1つの方法は複数ファイルをアーカイブして、1つのファイルにしてPUSHする方法で、ツールとしてはzip,lha,tarと呼ばれるようなアーカイブツールがある。

また、メール伝送に複数ファイルを添付する方法や、httpのアップロードに使われるformのpostで送り側のcgiに対しmultipartで送るといった方法がある。

専用のプロトコル（ソケット通信等）を設計して、1回のセッションで複数データを送る方法も考えられる。

受信側では、複数データのうち、どのデータが親であるか判断できなければならない。これには下記のような方法がある。

親ファイルだと判別できるように、ファイル名や拡張子に規則を持たせる。

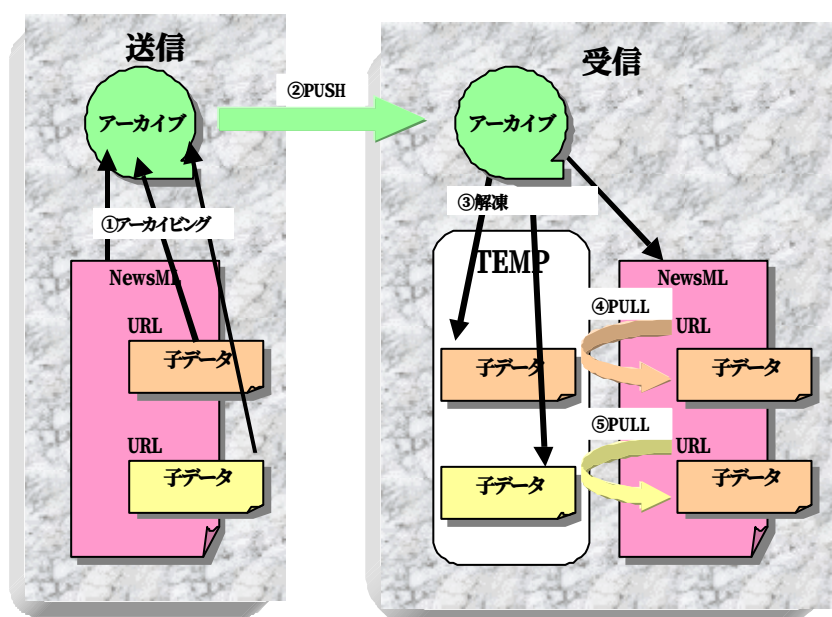
親ファイルを、特別なフォルダのパス付きで格納する。

親ファイルのファイル名を記述した固定ファイル名のファイルを同胞する。

他の手段でファイル名を伝える。

送り手は、データ中のURLに、プロトコルやアドレスを含まないファイルパスだけを記述することになる。

受け手は、親ファイルのURLを参照して子ファイルの取り込みを行う。



利用シーンと特徴

送り手は、あらかじめ親子関係などを再帰的に分析して必要な子ファイル把握する必要がある。

一度の通信で複数のデータを伝送することができる。

PUSH のみで送るため、常時接続でない場合も利用できる。

アーカイブの場合解凍しないと中身がどのようなデータか判断できない。

アーカイバーは圧縮を行うため伝送効率が良くなる。

添付資料-DefaultVocabularyFor の Context 属性

DefaultVocabularyFor/@Context の適用範囲（スコープ）

DefaultVocabularyFor/@Context の適用範囲は、Catalog 要素の親要素を基点として、そのサブツリーが、適用範囲（スコープ）になる。これは、すべてにおいて最優先され、適用範囲（スコープ）外に影響することはない。

DefaultVocabularyFor/@Context の値は何を示すか？

DefaultVocabularyFor/@Context の値を XPATH として評価する。適用範囲（スコープ）内でこの XPATH に一致する属性に、省略時のコントロールド・ボキャブラリが提供される。もし、Context 属性の値が属性を示していないとき、"/@FormalName"が省略されていることを表す。

たとえば、下記の場合

```
<NewsComponent>
```

```
<NewsComponent>
```

```
<Catalog>
```

```
<Resource>
```

```
<Url>topicset.ipc-format.xml</Url>
```

```
<DefaultVocabularyFor Context="/NewsML//Format"/> ...
```

```
...
```

```
</Catalog>
```

```
<ContentItem Duid="CI-001">
```

```
<Format FormalName="..."/>
```

```
...
```

```
</ContentItem>
```

```
</NewsComponent>
```

```
<NewsComponent>
```

```
<ContentItem Duid="CI-002">
```

```
<Format FormalName="..."/>
```

```
...
```

```
</ContentItem>
```

```
</NewsComponent>
```

<Catalog>の親要素(この例の場合 ContentItem Duid="CI-001")を基点として Context の文字列を XPATH として一致する属性が対象となる。Context 属性の値の書き方に関わらず、Duid="CI-001"の ContentItem 内にある Resourceは、Duid="CI-002"の ContentItem に影響を及ぼすことはない。したがって、Context="/NewsML//Format"のような記述は無意味であるために、そのような表現をすることを推奨しない。

IPTC NewsML 仕様書 V1.00 と同じ効果を表す記述方法は、Context 属性に"/要素名"もしくは、"/要素名/@FormalName"と記述すれば XPATH と適用範囲（スコープ）が一致するので、理想的である。

参考: <DefaultVocabularyFor Context="Format"/>の表す意味

つぎのような記述があったとき、

```
<NewsComponent>
  <Catalog>
    <Resource>
      <Url>topicset.iptc-format.xml</Url>
      <DefaultVocabularyFor Context="Format"/>
      ...
    </Catalog>
    <ContentItem Duid="CI-001">
      <Format FormalName="..." />
      ...
    </ContentItem>
  </NewsComponent>
```

DefaultVocabularyFor の
適用範囲 (スコープ)

IPTC NewsML 仕様書 V1.00 では、Catalog 要素の親要素のサブツリーにある、すべての要素に対する Format の FormalName 属性についての省略時コントロール・ボキャブラリを提供することを表した。

IPTC NewsML 仕様書 V1.02 からは Catalog 要素の親要素を基点とした XPATH になるので、Context="Format" のように、"/" がいない場合、NewsComponent 直下の要素を表すことになる。NewsComponent 直下に Format 要素は存在しないので、この指定では、どの Format 要素にも適用されないので注意が必要である。

DefaultVocabularyFor/@Context の優先順位

もし、階層レベルの異なる Catalog/Resource/DefaultVocabularyFor から、XPATH で一致する要素があった場合、当該要素に近い親階層の Catalog による指定が優先される。もし、この Catalog/Resource のコントロール・ボキャブラリに一致する FormalName と Scheme がなかった場合、XPATH が一致するさらに親階層の Catalog/Resource から一致するコントロール・ボキャブラリを探す。下図の場合、①、②の順でコントロール・ボキャブラリから一致する FormalName と Scheme を探すことになる。

```
<NewsComponent>
  <Catalog>
    <Resource>
      <Url>topicset.iptc-format.xml</Url>
      <DefaultVocabularyFor Scheme="IptcFormat" Context="..//Format"/>
      ...
    </Catalog>
    <NewsComponent>
      <Catalog>
        <Resource>
          <Url>topicset.iptc-format.xml</Url>
          <DefaultVocabularyFor Scheme="NskFormat" Context="..//Format"/>
          ...
        </Catalog>
        <ContentItem Duid="CI-001">
          <Format FormalName="..." />
          ...
        </ContentItem>
      </NewsComponent>
    </Catalog>
  </NewsComponent>
```

DefaultVocabularyFor の
適用範囲 (スコープ)

②

DefaultVocabularyFor の
適用範囲 (スコープ)

①

```
</ContentItem>
</NewsComponent>
```

DefaultVocabularyFor 要素が複数連続して記述されている場合

一つの Catalog/Resource 要素のなかに複数の DefaultVocabularyFor 要素があり、Context 属性の値も同じ場合がある。このような場合、省略時 Scheme が複数提供されることになる。

```
<Catalog>
  <Resource>
    <Url>topicset.iptc-format.xml</Url>
    <DefaultVocabularyFor Scheme="IptcFormat" Context="//Format"/>
    <DefaultVocabularyFor Scheme="NskFormat" Context="//Format"/>
  </Resource>
</Catalog>
```

上記の場合、Format 要素の FormalName 属性にコントロールド・ボキャブラリが適用されるが、2種類の省略時 Scheme が提供される。たとえばコントロールド・ボキャブラリから、先に Scheme が“IptcFormat”で、一致する FormalName を探し、見つからなければ Scheme が“NskFormat”で一致する FormalName を探す。

どちらの Scheme を先に探すかは、明確な規定がないこともあり、探す順序によって結果が異なるコントロールド・ボキャブラリを作成することは推奨しない。

同じように DefaltVocabularyFor 要素が連続していても、Context 属性が異なる場合もある。この場合は、複数の要素に同じコントロールド・ボキャブラリが使われることを意図している。このような場合は、たいてい Scheme 属性が異なり、Context に記述される該当 XPATH 要素ごとに Scheme を変えることによってコントロールド・ボキャブラリの候補を絞り込むような工夫がされている。

```
<TopicSet Duid="LocalTopicSet" Scheme="IptcTopicType" FormalName="LocalVocabulary" >
  <Topic Duid="BooleanTrue">
    <TopicType Scheme="IptcTopicType" FormalName="Property"/>
    <FormalName Scheme="Boolean">True</FormalName>
  </Topic>
  <Topic Duid="BooleanFalse">
    <TopicType Scheme="IptcProperty" FormalName="Property"/>
    <FormalName Scheme="Boolean">False</FormalName>
  </Topic>
  <Topic Duid="DayMonday">
    <TopicType Scheme="IptcProperty" FormalName="Property"/>
    <FormalName Scheme="Day">Monday</FormalName>
  </Topic>
  <Topic Duid="DayTuesday">
    <TopicType Scheme="IptcProperty" FormalName="Property"/>
    <FormalName Scheme="Day">Tuesday</FormalName>
  </Topic>
</TopicSet>
```

```
...
<Catalog>
  <Resource>
    <Url>#LocalTopicSet</Url>
    <DefaultVocabularyFor Scheme="Boolean" Context="Property[@FormalName='Print']/@Value"/>
    <DefaultVocabularyFor Scheme="Day" Context="Property[@FormalName='PrintDay']/@Value"/>
  </Resource>
</Catalog>
...
<Property FormalName="Print" Value="True"/>
<Property FormalName="PrintDay" Value="Monday"/>
```

第4期 NewsML チーム委員 名簿

◎＝幹事、○＝副幹事（敬称略）

	社名	職名	氏名
	朝日新聞東京本社	製作本部東京制作 2部コントローラ	荒井 一幸
	〃	製作本部東京システム 1部	内藤 和明
○	毎日新聞東京本社	制作技術局技術センター制作システムグループ技師	田中 孝次
	〃	制作技術局技術センター制作システムグループ	塚田 重博
	〃	制作技術局技術センターメディア 業務担当副部長	小野寺 尚希
	〃	制作技術局技術センターメディア 業務システムグループ	原 貴彦
◎	読売新聞東京本社	メディア戦略局編集部主任	三宅 学
	〃	メディア戦略局編集部	野口 潔
○	日本経済新聞社	情報技術本部システムリーダー	森口 剛
	東京新聞	編集局機報部長	杉本 雅昭
	〃	編集局機報部	藤田 俊広
○	〃	編集局機報部	加島 裕士
	〃	制作センターシステム運行部部次長	武江 完次
	産経新聞東京本社	制作局報道システム部次長	大島 義衛
	〃	制作局システム管制部	土屋 学
	〃	デジタルメディア局システム部	林 裕司
	日刊スポーツ新聞社	経営企画室	竹谷 耕司
	〃	情報システム本部システム部	山口 晃
	〃	情報システム本部システム部	中村 貴史
	スポーツニッポン新聞東京本社	システム本部開発・ネットワーク部技師補	荒井 雅樹
	東京ニュース通信社	情報システム室システム管理部課長	尾西 龍吾
	〃	情報システム室システム管理部	杉浦 英生
○	共同通信社	システム局システム企画室	篠塚 裕志
	〃	システム局システム管制部	溝口 博則
	〃	システム局システム開発部	内田 強
	時事通信社	システム開発局開発部主任	森本 佳輝
	〃	システム開発局開発部	赤倉 優蔵
	読売新聞大阪本社	メディア戦略室事業部	藤井 浩一
	産経新聞大阪本社	制作局報道システム部	小林 雅彦
	〃	制作局システム管制部	池田 正行
	スポーツニッポン新聞大阪本社	システム本部	大利 哲也
	〃	システム本部	志岐 忠彦

北海道新聞社	情報技術局技術開発部係長課長職	渡辺 邦彦
岩手日報社	制作局技術機報部	川又 淳
河北新報社	システム局システム部	五井 克浩
山形新聞社	制作局システム部	名取 信
福島民報社	総合メディア室メディア企画部	長谷川 満
山梨日日新聞社	編集局システム運用部	石川 秀和
静岡新聞社	制作技術局システム開発部	築地 宏和
信濃毎日新聞社	制作局電算部	山岸 拓巨
中日新聞社	編集局機報部	上 学
京都新聞社	メディア・システム局システム部主任	津浦 康男
神戸新聞社	製作本部次期システム準備室	園田 誠
〃	メディア局情報技術部	前川 雄樹
山陽新聞社	編成局システム部	武元 哲哉
〃	編成局システム部	平本 肇
徳島新聞社	メディア局情報部	河野 仁
西日本新聞社	システム技術局情報システム部次長	塩崎 真治
熊本日日新聞社	メディア開発局システム開発部次長	岡松 暁夫
南日本新聞社	メディア開発局システム開発部	松元 博幸
日刊編集センター	ラテ配信局マネージャー	一ノ瀬 智
日経ラジオ社	編成報道局チーフディレクター	掛原 雅行
イースト	コミュニケーション事業部シニアマネージャー	藤原 隆弘
産能コンサルティング	営業部報道システム担当主任	鈴木 辰也
東芝ソリューション	業種ソリューション第二部第1期ソリューション第二担当	淵山 央樹
ニコンシステム	第3システム本部第4開発部部長	宮本 亮
〃	第3システム本部第4開発部	山下 良一
日本アイ・ビー・エム	ソフトウェア開発研究所情報マネジメント技術部長 デジタルメディア担当	石井 宏和
〃	ソフトウェア開発研究所情報マネジメント技術第二開発アーキテクト	柴山 英俊
日本システム技術	営業部新聞システム営業統括部長	林 克美
〃	技術 1部	白鳥 和夫
〃	技術 3部	馬場 一行
日本電気	メディア・エネルギーソリューション事業部 ソリューションビジネス推進部マネージャ	佐藤 剛
〃	メディア・エネルギーソリューション事業部 ソリューションビジネス推進部主任	假谷 元
〃	第三公共システム開発事業部	大内 龍也
日本電気エンジニアリング	インターネットターミナル事業部ニュースメディア開発部	松江 則彦
富士通	情報メディア事業本部報道メディアシステム開発統括部	加藤 英明

//	情報メディア事業本部報道メディアシステム開発統括部	横田 雅人
//	情報メディア事業本部報道メディアシステム開発統括部	吉田 裕司
松下電送グラフィックプリンティング	システムエンジニアリング部システムパブリッシング課主任技師	宮本 和幸
//	システムエンジニアリング部システムパブリッシング課主任技師	石川 茂雄
//	システムエンジニアリング部システムパブリッシング課	西川 博己
ロココ	東京事業部システムコラボレーション部課長代理	高橋 和也
//	大阪事業部企画営業部企画課課長	上田 善行

NewsML サポートチーム 第1期委員 名簿

◎＝幹事、○＝副幹事 (敬称略)

	社名	職名	氏名
○	朝日新聞東京本社	製作本部東京制作 2部コントローラ	荒井 一幸
○	毎日新聞東京本社	制作技術局技術センター制作システムグループ技師	田中 孝次
	読売新聞東京本社	メディア戦略局編集部	野口 潔
	日本経済新聞社	情報技術本部主査	河村 俊介
◎	東京新聞	編集局機報部	加島 裕士
	産経新聞東京本社	制作局システム管理部	土屋 学
	報知新聞社	制作局システム部主査	小堀 弘治
	日刊スポーツ新聞社	システム局	中村 貴史
	スポーツニッポン新聞社	大阪本社システム局	大利 哲也
	東京ニュース通信社	情報システム室システム管理部	杉浦 英生
○	共同通信社	システム局システム開発部主任	三輪 雅信
○	時事通信社	システム開発局開発部	赤倉 優蔵
	読売新聞大阪本社	メディア戦略室事業部	藤井 浩一
	産経新聞大阪本社	制作局システム管理部	池田 正行
	朝日放送	放送技術局情報技術センター	赤藤 倫久
	北海道新聞社	情報技術局技術開発部	橋本 尚人
	岩手日報社	制作局技術機報部	川又 淳
	河北新報社	システム局システム部	五井 克浩
	山形新聞社	制作局システム部	名取 信
	福島民報社	総合メディア室メディア企画部	長谷川 満
	神奈川新聞社	デジタルメディア局システム部副主事	高橋 利明
	山梨日日新聞社	編集局システム運用部	石川 秀和
	静岡新聞社	制作技術局システム開発部	築地 宏和
	信濃毎日新聞社	技術局技術開発部	山岸 拓巨
	中日新聞社	編集局機報部	上 学
	新潟日報社	システム室システム部	大島 拓郎
	福井新聞社	メディア開発局システム開発部	吉田 秀樹
	京都新聞社	メディア・システム局システム部主任	津浦 康男
	神戸新聞社	メディア局情報技術部	前川 雄樹
	山陽新聞社	システム局システム部	平本 肇
	徳島新聞社	編集・システム総本部システム本部システム部係長	河野 仁
	西日本新聞社	システム技術局情報システム部副主事	高木 弘典
	熊本日日新聞社	メディア開発局システム開発部次長	岡松 暁夫

NewsML サポートチーム オブザーバー参加者名簿

社名	職名	氏名
毎日新聞東京本社	制作技術局技術センターメディア 業務担当副部長	小野寺 尚希
読売新聞東京本社	メディア戦略局編集部	宮脇 鉦介
東京新聞	制作センターシステム運行部	田中 信之
産経新聞東京本社	制作局報道システム部次長	大島 義衛
報知新聞社	制作局システム部	茂原 栄太
日刊スポーツ新聞社	システム局	山口 晃
スポーツニッポン新聞社	東京本社システム局開発・ネットワーク部	阿部 孝男
東京ニュース通信社	情報システム室システム管理部課長	尾西 龍吾
共同通信社	システム局システム管制部	溝口 博則
時事通信社	システム開発局開発部	立元 智幸
産経新聞大阪本社	制作局報道システム部	小林 雅彦
北海道新聞社	情報技術局システム開発本部技術開発部部次長	渡辺 邦彦
中日新聞社	システム統括本部システム統括部	山尾 信二
新潟日報社	システム室システム部主任	石川 尚彦
京都新聞社	メディア・システム局システム部課長同等	高乗 昌人
神戸新聞社	製作本部次期システム推進室	園田 誠
西日本新聞社	システム技術局情報システム部次長	塩崎 真治
日刊編集センター	ラテ配信局マネージャー	一ノ瀬 智
イースト	コミュニケーション事業部シニアマネージャー	藤原 隆弘
NTTデータ	メディア開発ビジネスユニット課長	佐藤 新市
産能コンサルティング	営業部報道システム担当主任	鈴木 辰也
東芝ソリューション	ソリューション第二事業部業種ソリューション第二部 新聞ソリューション第二担当主任	瀧山 央樹
ニコンシステム	第三システム本部第四開発部グループリーダ	山下 良一
日本アイ・ピー・エム	ソフトウェア開発研究所情報マネジメント技術主任	橋本 光治
日本システム技術	新聞メディア事業部営業統括	林 克美
日本電気	メディア・エネルギーソリューション事業部 ソリューションビジネス推進部マネージャ	佐藤 剛
NECエンジニアリング	インターネットターミナル事業部 第二メディアターミナル開発部	松江 則彦
パナソニックCCグラフィックス	システムエンジニアリンググループ新聞製作担当主任技師	宮本 和幸
富士通	情報メディア事業本部報道メディアシステム開発統括部	横田 雅人
富士フイルムグラフィックシステムズ	技術二部	横山 和夫

