



# Publishing Your Stuff With NewsML

**Anthony B. Coates**  
**Chief Technology Office, Reuters**

`tony.coates@reuters.com`

**XML DevCon Europe 2001**

## Overview

- » Aim of this presentation
- » What is NewsML
- » Basics of NewsML
- » Building a Website
- » Conclusion

## The Aim

- » This aim of this presentation is **not** to comprehensively explain NewsML
- » Instead, the aim is to show how to use just enough of NewsML to get started building a Website
- » This is just one possible approach
- » This is just a starting point

## What is NewsML

- » XML format for packaging multimedia news
- » Not a format for news content
- » Allows news content to be logically grouped into stories
- » Allows rich metadata to support searching, indexing, etc.

## What is NewsML

- » Conceived in **Reuters** (<http://about.reuters.com/>) in 1997
- » Open NewsML effort initiated through **IPTC** (<http://www.iptc.org/>) by Reuters in 1999
- » NewsML 1.0 approved by IPTC in Q4/2000

## NewsML Toolkit

- » Reuters & **WAVO** (<http://www.wavo.com/>) have funded a **NewsML toolkit** (<http://about.reuters.com/researchandstandards/firstcontact/newml-toolkit/>), currently Java, to help developers work with NewsML
- » Written by **David Megginson** (<http://www.megginson.com/>)
- » The toolkit will be used as an aid during this presentation

## NewsML Basics

- » This is a deliberately incomplete introduction to NewsML
- » Elements will only be covered where they are required to build a sample Website
- » For more details about NewsML, please see the NewsML 1.0 **Functional Specification & DTD** (<http://www.iptc.org>)

## NewsML Basics: NewsML, NewsItem

- » A NewsML document starts with a **<NewsML>** element
- » Can contain any number of **<NewsItem>** elements
- » Also must contain a **<NewsEnvelope>** element which must contain the date & time of transmission
- » Transmission is not a relevant concept here, so the **<NewsEnvelope>** will be quietly ignored
- » Even the smallest NewsML file contains a lot of fistful of metadata elements, so it really pays to use a DTD-sensitive XML editor

## NewsML Basics: Minimum NewsML

```
file #1: files/newsml-minimum.xml
<!DOCTYPE NewsML SYSTEM "NewsMLv1.0.dtd">
<NewsML>
  <NewsEnvelope>
    <DateAndTime>20010221T203000+0000</DateAndTime>
  </NewsEnvelope>
  <NewsItem>
    Minimum NewsItem Identification [1]
    Minimum NewsItem NewsManagement [2]
  </NewsItem>
</NewsML>
```

## NewsML Basics: Minimum NewsML

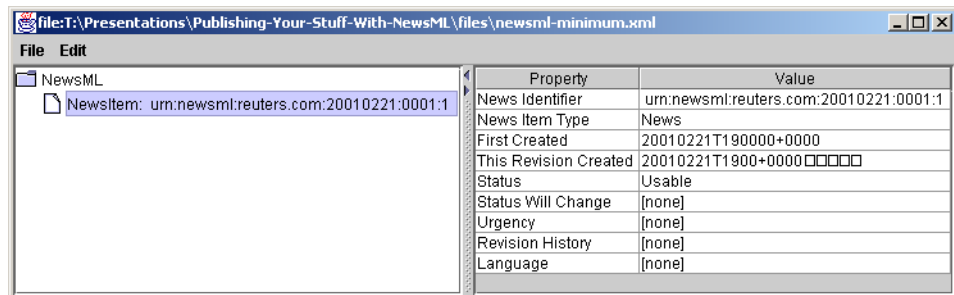
```
macro: Minimum NewsItem Identification [1]
<Identification>
  <NewsIdentifier>
    <ProviderId>reuters.com</ProviderId>
    <DateId>20010221</DateId>
    <NewsItemId>0001</NewsItemId>
    <RevisionId PreviousRevision = "0" Update =
      "N">1</RevisionId>
    <PublicIdentifier>
      urn:newsml:reuters.com:20010221:0001:1
    </PublicIdentifier>
  </NewsIdentifier>
</Identification>
Invoked in files #: 1
```

## NewsML Basics: Minimum NewsML

```
macro: Minimum NewsItem NewsManagement [2]
<NewsManagement>
  <NewsItemType FormalName = "News"/>
  <FirstCreated>20010221T190000+0000 </FirstCreated>
  <ThisRevisionCreated>20010221T1900+0000
  </ThisRevisionCreated>
  <Status FormalName = "Usable"/>
</NewsManagement>
Invoked in files #: 1
```

## NewsML Basics: NewsML Viewer

- » The NewsML Viewer (from the Reuters/WAVO NewsML Toolkit) hides metadata elements, so the minimal NewsML file looks like:



Property	Value
News Identifier	urn:newsml:reuters.com:20010221:0001:1
News Item Type	News
First Created	20010221T190000+0000
This Revision Created	20010221T1900+0000□□□□
Status	Usable
Status Will Change	[none]
Urgency	[none]
Revision History	[none]
Language	[none]

## NewsML Basics: NewsItem, NewsComponent, ContentItem

- » A **<NewsItem>** typically corresponds to a news story, or a group of news stories
- » A **<NewsItem>** typically contains 1 or more **<NewsComponent>** elements
- » A **<NewsComponent>** can contain **<ContentItem>**, **<NewsComponent>**, **<NewsItem>**, or **<NewsItemRef>** elements
- » A **<ContentItem>** is a raw component of a story
- » Could be text, an image, audio, video, etc.

## NewsML Basics: NewsComponent

- » NewsML stories can be built up from multiple items
- » Example: alternative main text in 3 languages + alternative images in 2 resolutions + alternative video with audio in 2 languages + background text and image for a sidebar in one language
- » In NewsML, a **<NewsComponent>** can group components/items as alternatives (*use one of these*) or complements (*use these together*)
- » A **<NewsComponent>** can also be marked as essential

## NewsML Basics: NewsComponent, NewsLines

- » A **<NewsComponent>** can have a **<NewsLines>** element
- » Within the **<NewsLines>** can be a headline, a byline, a dateline, copyright, keywords, etc.
- » So, in building Web pages from NewsML, the content of the pages must be structured using **<NewsItem>**, **<NewsComponent>**, **<ContentItem>**, **<NewsLines>**, and **<NewsItemRef>** elements
- » In addition, there will be elements required by the NewsML DTD

## Hello World

- » CS & IT examples usually start with something which prints “Hello World”
- » So, consider the following “Hello World” Web page:

## Hello World

```
file #2: files/hello-world.html
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>Hey, hello world!</p>
  </body>
</html>
```

## Hello World

- » What might an equivalent NewsML document look like?
- » One possibility is to use a **<NewsItem>** containing a **<NewsComponent>** with the following structure:

## Hello World

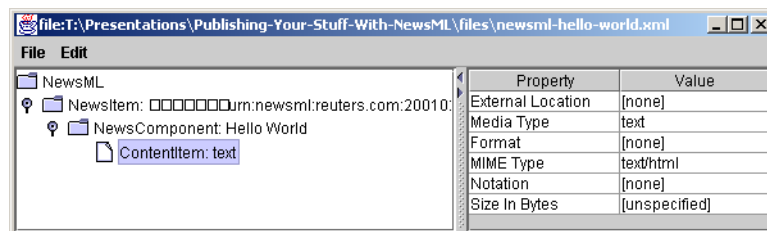
```
macro: Hello World NewsComponent [3]
<NewsComponent Essential = "no" EquivalentList = "no">
  <NewsLines>
    <HeadLine>Hello World</HeadLine>
  </NewsLines>
  Hello World ContentItem [4]
</NewsComponent>
Invoked in files #: 3
```

## Hello World

```
macro: Hello World ContentItem [4]
<ContentItem>
  <MediaType FormalName = "text"/>
  <MimeType FormalName = "text/html"/>
  <DataContent>
    <html>
      <body>
        <p>Hey, hello world!</p>
      </body>
    </html>
  </DataContent>
</ContentItem>
Invoked in macros #: 3
```

## Hello World

» Using the NewsML Viewer, this looks like



## Hello World

- » So what does this buy you?
- » Clearly, the NewsML version is much longer than the simple XHTML version
- » The answer is that you will not see gains where the content is this simple
- » You only start to win when you have enough **interconnected** page components and **metadata** to make it difficult to manage your XHTML as simple pages
- » This is when NewsML can give you the extra flexibility that you need

## Sample Web Site

- » In a 90 minute seminar, it isn't possible to give a demonstration of how to use NewsML to build an award-winning multimedia news site that would give major news organisations a run for their money
- » It is possible to show how a small interconnected Web site can be generated from the information in a **single NewsML file**
- » The sample site will contain Web pages about 6 countries, connected using metadata from the NewsML source

## Sample Web Site

- » The sample site is build around 9 Haiku (17 syllable poems)
- » Each Haiku connects one or more of 6 countries: U.S.A., U.K., Australia, France, Germany, Bulgaria
- » The following Haiku is about the U.S. alone:

## Sample Web Site: U.S. Haiku

```
macro: U.S. ContentItem [ 5 ]
<ContentItem>
  <MediaType FormalName = "text"/>
  <MimeType FormalName = "text/html"/>
  <DataContent>
    <html>
      <body>
        <pre>Donut. You go around without end. When
        you are gone, where is the hole?</pre>
      </body>
    </html>
  </DataContent>
</ContentItem>
Invoked in macros #: 6
```

## Sample Web Site: U.S. Haiku

- » The **<ContentItem>** is part of a **<NewsComponent>** which refers to the “US” Topic:

```
macro: U.S. NewsComponent [6]
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
  </DescriptiveMetadata>
  U.S. ContentItem [5]
</NewsComponent>
Invoked in files #: 4
```

## Sample Web Site: U.S. Haiku

- » The “US” topic is defined in a separate **<NewsItem>** which groups the topics (countries):

```
macro: U.S. Topic [7]
<Topic Duid = "US">
  <TopicType FormalName = "Country"/>
  <FormalName Scheme = "ISO-2">US</FormalName>
  <Description>America, the Home of the
  Brave</Description>
  <Property FormalName = "Capital" Value = "Washington DC"/>
</Topic>
Invoked in files #: 4
```

## Sample Web Site: Multiple Topics

- » The countries are connected by Haiku which refer to multiple topics (countries):

```
macro: Liberty DataContent [8]
<DataContent>
  <html>
    <body>
      <pre>A statue, a monument. Yet the sculptor
        fades into history.</pre>
    </body>
  </html>
</DataContent>
Invoked in macros #: 9
```

## Sample Web Site: Multiple Topics

```
macro: Liberty ContentItem [9]
<ContentItem>
  <MediaType FormalName = "text"/>
  <MimeType FormalName = "text/html"/>
  Liberty DataContent [8]
</ContentItem>
Invoked in macros #: 10
```

## Sample Web Site: Multiple Topics

```
macro: Liberty NewsComponent [10]
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
    <TopicOccurrence Topic = "#FR" HowPresent = "RelatesTo"/>
  </DescriptiveMetadata>
  Liberty ContentItem [9]
</NewsComponent>
Invoked in files #: 4
```

## Sample Web Site: Flags

» A **<NewsItem>** groups image URLs for the national flags:

```
macro: Australian Flag ContentItem [11]
<ContentItem Href = "images/AU.gif">
  <MediaType FormalName = "image"/>
  <MimeType FormalName = "image/gif"/>
</ContentItem>
Invoked in macros #: 12
```

## Sample Web Site: Flags

```
macro: Australian Flag NewsComponent [12]
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#AU"/>
  </DescriptiveMetadata>
  Australian Flag ContentItem [11]
</NewsComponent>
Invoked in files #: 4
```

## Sample Web Site: Anthems

- » Another **<NewsItem>** groups audio URLs for the national anthems:

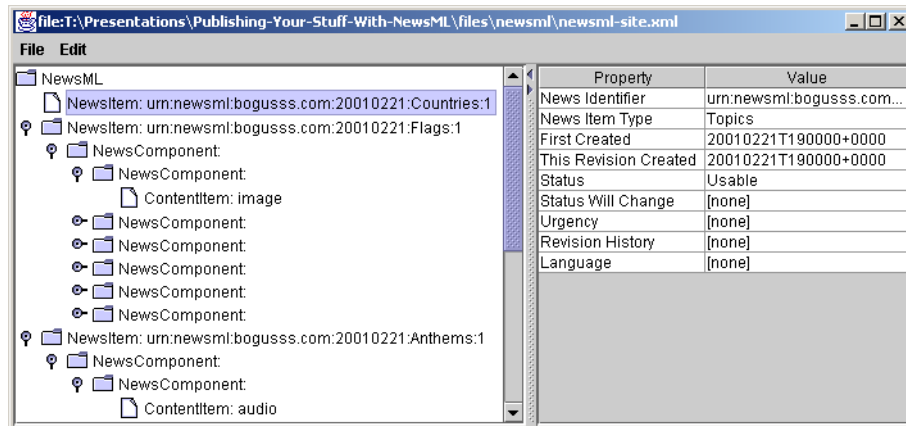
```
macro: Bulgarian Anthem ContentItem [13]
<ContentItem Href = "audio/BG.mid">
  <MediaType FormalName = "audio"/>
  <MimeType FormalName = "audio/midi"/>
</ContentItem>
Invoked in macros #: 14
```

## Sample Web Site: Anthems

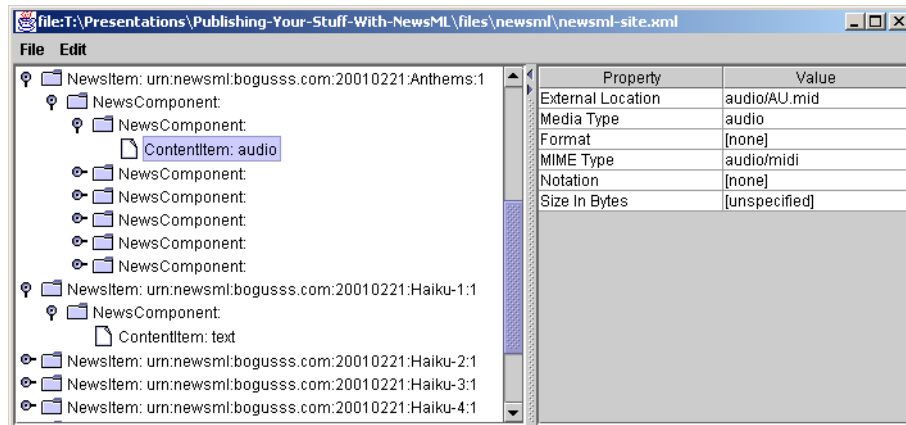
```

macro: Bulgarian Anthem NewsComponent [14]
<NewsComponent Essential = "no" EquivalentsList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#BG"/>
  </DescriptiveMetadata>
  Bulgarian Anthem ContentItem [13]
</NewsComponent>
Invoked in files #: 4
    
```

## Sample Web Site: Viewer



## Sample Web Site: Viewer



## Building the Website

- » To build the Web pages for the sample site, XSL-T will do nicely
- » Note that the XSL-T used here is specific to **Xalan** (<http://xml.apache.org/>) (Apache)
- » The offending item is the extension function **<redirect:write>**, used in generating multiple output files
- » XSL-T 1.0 does not define a standard way to generate multiple output files from a single stylesheet
- » That will change with XSL-T 1.1

## Building the Website: Index Page

- » The index page for the Web site will list the descriptions of the countries involved; these are defined in the topics:

```
macro: Australian Topic [15]
<Topic Duid = "AU">
  <TopicType FormalName = "Country"/>
  <FormalName Scheme = "ISO-2">AU</FormalName>
  <Description>Australia, the Land
  Downunder</Description>
  <Property FormalName = "Capital" Value = "Canberra"/>
</Topic>
Invoked in files #: 4
```

## Building the Website: Index Page

- » A suitable XPath to select the topics in the NewsML source follows. The matching topics are stored, as they are used multiple times:

```
macro: Store Topics [16]
<xsl:variable name = "topics" select =
  "//Topic[TopicType/@FormalName='Country']"/>
Invoked in files #: 5
```

## Building the Website: Index Page

- » The list of descriptions can be produced using

*macro: List All Descriptions [17]*

```
<ol>
  <xsl:for-each select = "$topics">
    <xsl:variable name = "topicFile">
      <xsl:call-template name = "get-topic-filename"/>
    </xsl:variable>
    List One Description [18]
  </xsl:for-each>
</ol>
Invoked in macros #: 19
```

## Building the Website: Index Page

- » Within each **<Topic>**, the text of the description is contained in its **<Description>**:

*macro: List One Description [18]*

```
<li>
  <a href = "{$topicFile}">
    <xsl:value-of select = "Description" disable-output-escaping =
      "no"/>
  </a>
</li>
Invoked in macros #: 17
```

- » With that, the index page just needs some HTML wrapping

## Building the Website: Index Page

```
macro: Index Page [19]
<html>
  <head>
    <title>Haiku by Country</title>
  </head>
  <body>
    <h1>Haiku by Country</h1>
    List All Descriptions [17]
  </body>
</html>
Invoked in files #: 5
```

## Building the Website: Screenshot



## Building the Website: Topic Pages

- » The individual Web pages for each topic (country) can now be built

```
macro: Build Topic Pages [20]
<xsl:for-each select = "$topics">
  <xsl:call-template name = "generate-topic-file">
    <xsl:with-param name = "topics" select = "$topics"/>
  </xsl:call-template>
</xsl:for-each>
Invoked in files #: 5
```

## Building the Website: Topic Pages

- » In building each topic's page, the filename is constructed

```
macro: Topic Page Template [21]
<xsl:template name = "generate-topic-file">
  <xsl:param name = "topics"/>
  <xsl:variable name = "filename">
    <xsl:call-template name = "get-topic-filename"/>
  </xsl:variable>
  Write Topic File [22]
</xsl:template>
Invoked in files #: 5
```

- » The topic code is extracted, and the reference constructed

## Building the Website: Topic Pages

```
macro: Write Topic File [22]
<redirect:write file = "{$filename}">
  <xsl:variable name = "topicCode" select = "@Duid"/>
  <xsl:variable name = "topicRef">
    <xsl:text disable-output-escaping = "no">#</xsl:text>
    <xsl:value-of select = "$topicCode" disable-output-escaping =
      "no"/>
  </xsl:variable>
  Topic Page Template Body [31]
</redirect:write>
Invoked in macros #: 21
```

## Building the Website: Topic Flag

- » The flag URL is located by searching for a **<ContentItem>** which contains an image and has the matching topic:

```
macro: Locate Flag URL [23]
<xsl:variable name = "flag">
  <xsl:value-of select = "//NewsComponent/ContentItem[
    ../DescriptiveMetadata/TopicOccurrence/@Topic = $topicRef and
    MediaType/@FormalName = 'image']/@Href"
  disable-output-escaping = "no"/>
</xsl:variable>
Invoked in macros #: 31
```

## Building the Website: Topic Flag

- » The anthem URL is located by searching for a **<ContentItem>** which contains audio and has the matching topic:

```
macro: Locate Anthem URL [24]
<xsl:variable name = "anthem">
  <xsl:value-of select = "//NewsComponent/ContentItem[
    ../DescriptiveMetadata/TopicOccurrence/@Topic = $topicRef and
    MediaType/@FormalName = 'audio']/@Href"
    disable-output-escaping = "no"/>
</xsl:variable>
Invoked in macros #: 31
```

## Building the Website: Multimedia Flag

- » Having located the flag and anthem, the flag is displayed and hyperlinked to the anthem:

```
macro: Multimedia Flag [25]
<div style = "float:right;">
  <a href = "{$anthem}" target = "anthem">
    <img src = "{$flag}" width = "108" height = "54" alt = "Click for
    the anthem"/>
  </a>
</div>
Invoked in macros #: 31
```

## Building the Website: Topic Haiku

» The Haiku for each topic are simply extracted:

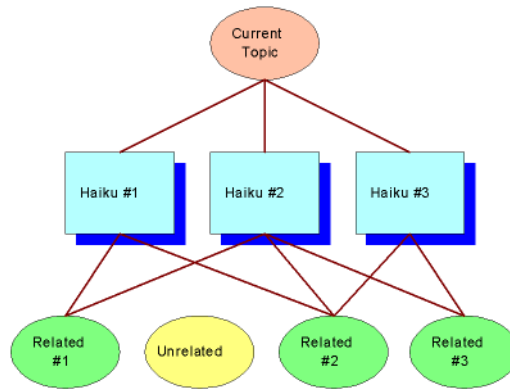
```
macro: Extract Haiku [26]
<xsl:for-each select =
  "//NewsItem[NewsManagement/NewsItemType/@FormalName =
  'Poetry']">
  <xsl:if test =
    "NewsComponent/DescriptiveMetadata/TopicOccurrence[
    @Topic = $topicRef]">
    Format Haiku [27]
    <hr/>
  </xsl:if>
</xsl:for-each>
Invoked in macros #: 31
```

## Building the Website: Topic Haiku

```
macro: Format Haiku [27]
<pre>
  <xsl:value-of select =
    "NewsComponent/ContentItem/DataContent/html/body/pre"
    disable-output-escaping = "no"/>
</pre>
Invoked in macros #: 26
```

## Building the Website: Related Topics

- » Related topics are found by searching checking the topics of all matching Haiku:



## Building the Website: Related Topics

- » The simplest procedure is to iterate through the available topics
- » When a matching topic (country) is located, its flag is displayed
- » The flag is hyperlinked to the page of that topic (country)

```
macro: List Related Topics [28]
<xsl:for-each select = "$topics">
  Generate thisTopicCode & thisTopicRef [33]
  Generate Flag & Page Links [34]
  Display Flag if Related [29]
</xsl:for-each>
Invoked in macros #: 31
```

## Building the Website: Related Topics

```
macro: Display Flag if Related [29]
<xsl:if test =
  "//NewsItem[NewsManagement/NewsItemType/@FormalName =
  'Poetry' and
  NewsComponent/DescriptiveMetadata/TopicOccurrence/@Topic=$topicRef
  and
  NewsComponent/DescriptiveMetadata/TopicOccurrence/@Topic=$thisTopicRef
  ]">
  Display Related Flag [30]
</xsl:if>
Invoked in macros #: 28
```

## Building the Website: Related Topics

```
macro: Display Related Flag [30]
<p>
  <a href = "{$thisTopicFile}">
    <img src = "{$thisFlag}" width = "54" height = "27" alt =
    "Related Flag"/>
  </a>
</p>
Invoked in macros #: 29
```

- » That is everything we need
- » Once the XSL-T script is run on the NewsML source, the final result can be viewed

## Building the Website: Topic Pages

Flags and anthems are copyright [thenationalanthems.com](http://thenationalanthems.com)  
Copyright details available at the website



[\[Contents\]](#)

### The United Kingdom, Home of Great Traditions

---

#### Related Topics



#### Haiku

---

Colony, they say.  
Are those old bonds so tight  
that they can never break?

---

## Building the Website: Topic Pages

Flags and anthems are copyright [thenationalanthems.com](http://thenationalanthems.com)  
Copyright details available at the website



[\[Contents\]](#)

### Germany, Because Common Sense is Uncommonly Fun

---

#### Related Topics



#### Haiku

---

Maginot.  
Not a wall, but a belief,  
that a shield can allay fear.

---

From ashes of war,  
we reach for the stars.  
And yonder the moon doth call.

## Conclusion

- » You have been introduced to how multimedia resources can be **organised** using NewsML
- » I hope you will now understand enough of the basics of NewsML to be able to use it for your own purposes
- » There is a lot more to NewsML than what it has been possible to cover here
- » This was just a starting point

## Conclusion

- » Keep in mind that as your needs grow, you can and should use **multiple** NewsML files
- » NewsML documents can easily reference information in other NewsML documents
- » NewsML is also designed to be transmitted and updated, so you could use it as the basis of a **distributed** Web environment
- » That is a topic for another talk, however

## Conclusion

- » This presentation was written in XML and converted to PDF using XSL (XSL-T & XSL-FO)
- » Thanks to **RenderX** (<http://www.renderx.com/>) for working on their **XEP** XSL-FO engine over a weekend so that I could do this presentation
- » The **NewsML mailing list** (<http://www.egroups.com/group/newsml/>) is hosted at **eGroups** (<http://www.egroups.com/>); membership open to all
- » The **latest version of this presentation** (<http://about.reuters.com/researchandstandards/events/2001/02/xml-devcon-europe-newsml/>) will be available on the Web

# Annexe

NewsML equivalent of the “Hello World” HTML sample:

```
file #3: files/newsml-hello-world.xml
<!DOCTYPE NewsML SYSTEM "NewsMLv1.0.dtd" [
<!ELEMENT p (#PCDATA)>
<!ELEMENT body (p)*>
<!ELEMENT html (body)>
]><NewsML>
  <NewsEnvelope>
    <DateAndTime>20010221T203000+0000</DateAndTime>
  </NewsEnvelope>
  <NewsItem>
    <Identification>
      <NewsIdentifier>
        <ProviderId>reuters.com</ProviderId>
        <DateId>20010221</DateId>
        <NewsItemId>0001</NewsItemId>
        <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
        <PublicIdentifier>
          urn:newsml:reuters.com:20010221:0001:1</PublicIdentifier>
        </NewsIdentifier>
      </Identification>
      <NewsManagement>
        <NewsItemType FormalName = "WebPage"/>
        <FirstCreated>20010221T190000+0000</FirstCreated>
        <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
        <Status FormalName = "Usable"/>
      </NewsManagement>
      Hello World NewsComponent [3]
    </NewsItem>
  </NewsML>
```

The NewsML source for the Haiku Web site:

```
file #4: files/newsml/newsml-site.xml
<!DOCTYPE NewsML SYSTEM "../rtr-slides.dtd"><NewsML>
  <NewsEnvelope>
    <DateAndTime>20010221T203000+0000</DateAndTime>
  </NewsEnvelope>
  <NewsItem>
    <Identification>
```

```

<NewsIdentifier>
  <ProviderId>bogusss.com</ProviderId>
  <DateId>20010221</DateId>
  <NewsItemId>Countries</NewsItemId>
  <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
  <PublicIdentifier>urn:newsml:bogusss.com:20010221:Countries:1</PublicIdentifier>
</NewsIdentifier>
</Identification>
<NewsManagement>
  <NewsItemType FormalName = "Topics"/>
  <FirstCreated>20010221T190000+0000</FirstCreated>
  <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
  <Status FormalName = "Usable"/>
</NewsManagement>
<TopicSet FormalName = "Countries">
  Australian Topic [15]
  <Topic Duid = "UK">
    <TopicType FormalName = "Country"/>
    <FormalName Scheme = "ISO-2">UK</FormalName>
    <Description>The United Kingdom, Home of Great
      Traditions</Description>
    <Property FormalName = "Capital" Value = "London"/>
  </Topic>
  U.S. Topic [7]
  <Topic Duid = "FR">
    <TopicType FormalName = "Country"/>
    <FormalName Scheme = "ISO-2">FR</FormalName>
    <Description>France, Where Scientists and Engineers are
      Artists and Lovers</Description>
    <Property FormalName = "Capital" Value = "Paris"/>
  </Topic>
  <Topic Duid = "DE">
    <TopicType FormalName = "Country"/>
    <FormalName Scheme = "ISO-2">DE</FormalName>
    <Description>Germany, Because Common Sense is Uncommonly
      Fun</Description>
    <Property FormalName = "Capital" Value = "Berlin"/>
  </Topic>
  <Topic Duid = "BG">
    <TopicType FormalName = "Country"/>
    <FormalName Scheme = "ISO-2">BG</FormalName>
    <Description>Bulgaria, Europe's Oldest State</Description>
    <Property FormalName = "Capital" Value = "Sofia"/>
  </Topic>
</TopicSet>

```

```

</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DateId>20010221</DateId>
      <NewsItemId>Flags</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsml:bogusss.com:20010221:Flags:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Images"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    Australian Flag NewsComponent [12]
    <NewsComponent Essential = "no" EquivalentList = "no">
      <DescriptiveMetadata>
        <TopicOccurrence HowPresent = "RelatesTo" Topic = "#UK"/>
      </DescriptiveMetadata>
      <ContentItem Href = "images/UK.gif">
        <MediaType FormalName = "image"/>
        <MimeType FormalName = "image/gif"/>
      </ContentItem>
    </NewsComponent>
    <NewsComponent Essential = "no" EquivalentList = "no">
      <DescriptiveMetadata>
        <TopicOccurrence HowPresent = "RelatesTo" Topic = "#US"/>
      </DescriptiveMetadata>
      <ContentItem Href = "images/US.gif">
        <MediaType FormalName = "image"/>
        <MimeType FormalName = "image/gif"/>
      </ContentItem>
    </NewsComponent>
    <NewsComponent Essential = "no" EquivalentList = "no">
      <DescriptiveMetadata>
        <TopicOccurrence HowPresent = "RelatesTo" Topic = "#FR"/>
      </DescriptiveMetadata>
      <ContentItem Href = "images/FR.gif">
        <MediaType FormalName = "image"/>
        <MimeType FormalName = "image/gif"/>
      </ContentItem>
    </NewsComponent>
  </NewsComponent Essential = "no" EquivalentList = "no">

```

```

</NewsComponent>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#DE"/>
  </DescriptiveMetadata>
  <ContentItem Href = "images/DE.gif">
    <MediaType FormalName = "image"/>
    <MimeType FormalName = "image/gif"/>
  </ContentItem>
</NewsComponent>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#BG"/>
  </DescriptiveMetadata>
  <ContentItem Href = "images/BG.gif">
    <MediaType FormalName = "image"/>
    <MimeType FormalName = "image/gif"/>
  </ContentItem>
</NewsComponent>
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DatId>20010221</DatId>
      <NewsItemId>Anthems</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Anthems:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Images"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    <NewsComponent Essential = "no" EquivalentList = "no">
      <DescriptiveMetadata>
        <TopicOccurrence HowPresent = "RelatesTo" Topic = "#AU"/>
      </DescriptiveMetadata>
      <ContentItem Href = "audio/AU.mid">
        <MediaType FormalName = "audio"/>
        <MimeType FormalName = "audio/midi"/>
      </ContentItem>
    </NewsComponent>
  </NewsComponent>

```

```

    </ContentItem>
</NewsComponent>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#UK"/>
  </DescriptiveMetadata>
  <ContentItem Href = "audio/UK.mid">
    <MediaType FormalName = "audio"/>
    <MimeType FormalName = "audio/midi"/>
  </ContentItem>
</NewsComponent>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#US"/>
  </DescriptiveMetadata>
  <ContentItem Href = "audio/US.mid">
    <MediaType FormalName = "audio"/>
    <MimeType FormalName = "audio/midi"/>
  </ContentItem>
</NewsComponent>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#FR"/>
  </DescriptiveMetadata>
  <ContentItem Href = "audio/FR.mid">
    <MediaType FormalName = "audio"/>
    <MimeType FormalName = "audio/midi"/>
  </ContentItem>
</NewsComponent>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence HowPresent = "RelatesTo" Topic = "#DE"/>
  </DescriptiveMetadata>
  <ContentItem Href = "audio/DE.mid">
    <MediaType FormalName = "audio"/>
    <MimeType FormalName = "audio/midi"/>
  </ContentItem>
</NewsComponent>
Bulgarian Anthem NewsComponent [14]
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>

```

```

    <DateId>20010221</DateId>
    <NewsItemId>Haiku-1</NewsItemId>
    <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
    <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-1:1</PublicIdentifier>
  </NewsIdentifier>
</Identification>
<NewsManagement>
  <NewsItemType FormalName = "Poetry"/>
  <FirstCreated>20010221T190000+0000</FirstCreated>
  <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
  <Status FormalName = "Usable"/>
</NewsManagement>
U.S. NewsComponent [6]
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DateId>20010221</DateId>
      <NewsItemId>Haiku-2</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-2:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    <DescriptiveMetadata>
      <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#AU" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#UK" HowPresent = "RelatesTo"/>
    </DescriptiveMetadata>
    <ContentItem>
      <MediaType FormalName = "text"/>
      <MimeType FormalName = "text/html"/>
      <DataContent>
        <html>
          <body>
            <pre>Colony, they say. Are those old bonds so tight
              that they can never break?</pre>
          </body>
        </html>
      </DataContent>
    </ContentItem>
  </NewsComponent>
</NewsItem>

```

```

    </html>
  </DataContent>
</ContentItem>
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DatId>20010221</DatId>
      <NewsItemId>Haiku-3</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-3:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  Liberty NewsComponent [10]
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DatId>20010221</DatId>
      <NewsItemId>Haiku-4</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-4:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    <DescriptiveMetadata>
      <TopicOccurrence Topic = "#AU" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#BG" HowPresent = "RelatesTo"/>
    </DescriptiveMetadata>
  </ContentItem>

```

```

<MediaType FormalName = "text"/>
<MimeType FormalName = "text/html"/>
<DataContent>
  <html>
    <body>
      <pre>One young, one old, and poles apart. Yet all
        can be bridged, as man and wife.</pre>
    </body>
  </html>
</DataContent>
</ContentItem>
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DateId>20010221</DateId>
      <NewsItemId>Haiku-5</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsml:bogusss.com:20010221:Haiku-5:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    <DescriptiveMetadata>
      <TopicOccurrence Topic = "#FR" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#DE" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
    </DescriptiveMetadata>
    <ContentItem>
      <MediaType FormalName = "text"/>
      <MimeType FormalName = "text/html"/>
      <DataContent>
        <html>
          <body>
            <pre>Maginot. Not a wall, but a belief, that a
              shield can allay fear.</pre>
          </body>
        </html>
      </DataContent>
    </ContentItem>
  </NewsComponent>
</NewsItem>

```

```

        </DataContent>
    </ContentItem>
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DateId>20010221</DateId>
      <NewsItemId>Haiku-6</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsml:bogusss.com:20010221:Haiku-6:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    <DescriptiveMetadata>
      <TopicOccurrence Topic = "#AU" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#FR" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
    </DescriptiveMetadata>
    <ContentItem>
      <MediaType FormalName = "text"/>
      <MimeType FormalName = "text/html"/>
      <DataContent>
        <html>
          <body>
            <pre>Your nuclear test, my nuclear family. Which
              then should come first?</pre>
          </body>
        </html>
      </DataContent>
    </ContentItem>
  </NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DateId>20010221</DateId>

```

```

    <NewsItemId>Haiku-7</NewsItemId>
    <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
    <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-7:1</PublicIdentifier>
  </NewsIdentifier>
</Identification>
<NewsManagement>
  <NewsItemType FormalName = "Poetry"/>
  <FirstCreated>20010221T190000+0000</FirstCreated>
  <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
  <Status FormalName = "Usable"/>
</NewsManagement>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence Topic = "#AU" HowPresent = "RelatesTo"/>
    <TopicOccurrence Topic = "#UK" HowPresent = "RelatesTo"/>
    <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
  </DescriptiveMetadata>
  <ContentItem>
    <MediaType FormalName = "text"/>
    <MimeType FormalName = "text/html"/>
    <DataContent>
      <html>
        <body>
          <pre>Our language divides us, but not so much as it
            unites us. Come talk.</pre>
        </body>
      </html>
    </DataContent>
  </ContentItem>
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DatId>20010221</DatId>
      <NewsItemId>Haiku-8</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-8:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>

```

```

    <Status FormalName = "Usable"/>
</NewsManagement>
<NewsComponent Essential = "no" EquivalentList = "no">
  <DescriptiveMetadata>
    <TopicOccurrence Topic = "#DE" HowPresent = "RelatesTo"/>
    <TopicOccurrence Topic = "#US" HowPresent = "RelatesTo"/>
  </DescriptiveMetadata>
  <ContentItem>
    <MediaType FormalName = "text"/>
    <MimeType FormalName = "text/html"/>
    <DataContent>
      <html>
        <body>
          <pre>From ashes of war, we reach for the stars. And
            yonder the moon doth call.</pre>
        </body>
      </html>
    </DataContent>
  </ContentItem>
</NewsComponent>
</NewsItem>
<NewsItem>
  <Identification>
    <NewsIdentifier>
      <ProviderId>bogusss.com</ProviderId>
      <DatId>20010221</DatId>
      <NewsItemId>Haiku-9</NewsItemId>
      <RevisionId PreviousRevision = "0" Update = "N">1</RevisionId>
      <PublicIdentifier>urn:newsm1:bogusss.com:20010221:Haiku-9:1</PublicIdentifier>
    </NewsIdentifier>
  </Identification>
  <NewsManagement>
    <NewsItemType FormalName = "Poetry"/>
    <FirstCreated>20010221T190000+0000</FirstCreated>
    <ThisRevisionCreated>20010221T190000+0000</ThisRevisionCreated>
    <Status FormalName = "Usable"/>
  </NewsManagement>
  <NewsComponent Essential = "no" EquivalentList = "no">
    <DescriptiveMetadata>
      <TopicOccurrence Topic = "#DE" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#FR" HowPresent = "RelatesTo"/>
      <TopicOccurrence Topic = "#BG" HowPresent = "RelatesTo"/>
    </DescriptiveMetadata>
    <ContentItem>
      <MediaType FormalName = "text"/>

```

```

<MimeType FormalName = "text/html"/>
<DataContent>
  <html>
    <body>
      <pre>Our currencies are tied, but not our hands.
      Can we not all share in growth?</pre>
    </body>
  </html>
</DataContent>
</ContentItem>
</NewsComponent>
</NewsItem>
</NewsML>

```

The XSL-T stylesheet used to create the Haiku Web site from “newsml-site.xml”:

```

file #5: files/xslt/newsml-to-website.xsl
<xsl:stylesheet version = "1.0" extension-element-prefixes = "redirect">
  <xsl:output method = "html"/>
  <xsl:template match = "/">
    Store Topics [16]
    <xsl:call-template name = "generate-index-file">
      <xsl:with-param name = "topics" select = "$topics"/>
    </xsl:call-template>
    Build Topic Pages [20]
  </xsl:template>
  <xsl:template name = "generate-index-file">
    <xsl:param name = "topics"/>
    <redirect:write file = "index.html">
      Index Page [19]
    </redirect:write>
  </xsl:template>
  Topic Page Template [21]
  <xsl:template name = "get-topic-filename">
    <xsl:value-of select = "@Duid" disable-output-escaping = "no"/>
    <xsl:text disable-output-escaping = "no">.html</xsl:text>
  </xsl:template>
</xsl:stylesheet>

```

The template for the Haiku Web site's topic (country) pages:

```

macro: Topic Page Template Body [31]
  Locate Flag URL [23]

```

```

    Locate Anthem URL [24]
<html>
  Topic Page Head [32]
  <body>
    <div>
      Multimedia Flag [25]
      Copyright Notice [35]
      <h5>
        <a href = "index.html">[ Contents ]</a>
      </h5>
      <h3 style = "text-align: center;">
        <xsl:for-each select = "$topics[@Duid = $topicCode]">
          <xsl:value-of select = "Description" disable-output-escaping = "no"/>
        </xsl:for-each>
      </h3>
      <hr/>
    </div>
    <div>
      <div style = "float:left; width:50%;">
        <h4>Related Topics</h4>
        List Related Topics [28]
      </div>
      <div style = "float:right; width:50%;">
        <h3>Haiku</h3>
        <hr/>
        Extract Haiku [26]
      </div>
    </div>
  </body>
</html>
  Invoked in macros #: 22

```

The macro used to create the <head> elements in the Haiku Web site's topic (country) pages:

```

macro: Topic Page Head [32]
<head>
  <title>
    <xsl:value-of select = "$topicCode" disable-output-escaping = "no"/>
    <xsl:text disable-output-escaping = "no"> Haiku</xsl:text>
  </title>
</head>
  Invoked in macros #: 31

```

Macro used to store the next topic code and reference when searching for related topics:

```
macro: Generate thisTopicCode & thisTopicRef [33]
<xsl:variable name = "thisTopicCode" select = "@Duid"/><xsl:variable name =
"thisTopicRef">
  <xsl:text disable-output-escaping = "no">#</xsl:text>
  <xsl:value-of select = "$thisTopicCode" disable-output-escaping = "no"/>
</xsl:variable>
Invoked in macros #: 28
```

Stores the URLs for the Web page and flag image for a particular topic (country):

```
macro: Generate Flag & Page Links [34]
<xsl:variable name = "thisFlag">
  <xsl:value-of select =
  "//NewsComponent/ContentItem[../DescriptiveMetadata/TopicOccurrence/@Topic
= $thisTopicRef and MimeType/@FormalName = 'image/gif']/@Href"
  disable-output-escaping = "no"/>
</xsl:variable><xsl:variable name = "thisTopicFile">
  <xsl:value-of select = "@Duid" disable-output-escaping = "no"/>
  <xsl:text disable-output-escaping = "no">.html</xsl:text>
</xsl:variable>
Invoked in macros #: 28
```

Copyright notice for the flag images and anthems:

```
macro: Copyright Notice [35]
<h5>Flags and anthems are copyright
<a href = "http://www.thenationalanthems.com/" target =
"copyright">thenationalanthems.com</a><br/>Copyright details
available
at the website</h5>
Invoked in macros #: 31
```