Enhanced "enclosures" support in RSS and ATOM Syndication

Vadim Zaliva, lord@crocodile.org
Alexander Sova, bird@noir.crocodile.org

December 15, 2004

Abstract

This document presents an extension of "enclosure" support in RSS and ATOM syndication formats.

This document's status is "draft". This is a proposal to the internet community in order to facilitate further discussion.

Contents

1 Acknowledgments

This work is based upon the previous work of many people. Specifically, we would like to mention Dave Winer, original author of "enclosure" element in RSS 2.0 specification[5]. We have also borrowed many concepts and ideas from a variety of internet standards and proposals. Most of our sources are listed on page 14.

Many people provided valuable comments and suggestions. We would like to mention Suzan Foster, Reto Bachmann-Gmuer, Danny Ayers.

2 Current Status or RSS enclosures

Recently RSS format began to be used for the distribution of audio content (known as "podcasting"). There are some emerging projects which consider the use of RSS/ATOM feeds for the distribution of video and other media.

RSS 2.0[5] defines an "enclosure" element which is already supported by several RSS agregators and feed generators.

2.1 Problems

In the course of using the "enclosure" element of RSS 2.0, several shortcoming were reported:

- It is only defined for RSS 2.0. There is no formal definition of similar element for RSS 1.0 and ATOM.
- It only allows simple flat lists of enclosures. It does not allow one to specify alternative formats, group enclosures, or specify relationships between them.
- It does not support any kind of non-abmbigous identification of individiual enclosures for the purpose of referencing them.
- It does not contain any means of specifying relationships between syndicated content and enclosure.
- It does not allow the specification of multiple alternative download locations.
- It does not allow the specification of download locations using file sharing protocols like BitTorrent.

2.2 Goals

In this document we will propose new, enhanced syntax for RSS "enclosures" which will address problems with the current "enclosure" element.

This new syntax should not conflict with the exiting one. Thus, RSS feed will be capable of containing both old and new-style enclosure definitions.

It should also be simple to implement, and be defined for both 1.0 and 2.0 versions of RSS and ATOM.

We will attempt to model it on existing, proven internet standards.

3 Enhanced Enclosures

3.1 Enclosures Structure

We will distiguish here between *Simple* and *Composite* enclosures. *Simple* enclosure will define a single external element, while *Composite* enclosure will have an internal structure and will be capable of containing other *Simple* or *Composite* elements.

An RSS "item" or ATOM "entry" element could contain multiple *Simple* or *Composite* enclosure elements. If more than one such element is included under either "item" or "atom:entry", then they are considered to be part of an implied *Composite* enclosure with kind "mixed" (see section 3.3).

All enclosure elements are defined in separate namespace. We will use 2 namepsaces: one for RSS 2.0 enclosure syntax and another for for RSS 1.0. Respective namsepace defintions are:

```
xmlns:enc="http://crocodile.org/ns/rss/2.0/enclosures"
xmlns:enc1="http://crocodile.org/ns/rss/1.0/enclosures"
```

3.2 Anatomy of "Simple" enclosure

Simple enclosures use the "Enclosure" element name and have the following attributes:

Attribute	Mandatory	Description
url	mandatory	URL of a file represented by this enclosure
type	mandatory	Media Type[9] of a data
transport	mandatory	Specifies what transport should be used to download content. (see section 3.5 for details)
location	optional	Spatial location of enclosure source (see section 3.6 for details)
length	optional	Content length of the data
rel	optional	This attribute describes the relationship of the enclosing element ("item", "atom:entry" or "Enclosures") to the anchor specified by the url attribute. The value of this attribute is a space-separated list of link types.
quality	optional	Relative quality factor. Quality factors allow the user or user agent to indicate the relative degree of preference for that media, using the value scale from 0 to 1 (see section 3.4 for details). The default value is 1.
Content-ID	optional	Unique identifier of this enclosure (must be unique within "item" or "atom:entry"). ("cid" URL schema[11] could be used).
description	optional	Human-readable brief description of the enclosure contents.

Simple enclosure is not allowed to have sub-elements.

3.3 Anatomy of "Composite" enclosure

Composite enclosures use the "Enclosures" (plural) element name and have the following attributes:

Attribute	Mandatory	Description
kind	optional	*kind* of enclosure - specifies the relationship between sub-elements (see below)
rel	optional	This attribute describes the relationship of the enclosing element ("item", "atom:entry" or another "Enclosures") to this enclosure. The value of this attribute is a space-separated list of link types.
Content-ID	optional	Unique identifier of this enclosure (must be unique within "item" or "atom:entry"). ("cid" URL schema[11] could be used).
description	optional	Human-readable brief description of the enclosure contents

Composite enclosures should have one or more nested “Enclosure” or “Enclosures” elements.

The “kind” attribute specifies the relationship between sub-elements of this composite enclosure. Defined values are:

parallel: This value recommends the display of all sub-elements simultaneously on both hardware and software that are capable of doing so. However, composing agents should be aware that some readers may lack this capability and will therefor show the sub-elements serially instead.

mixed: This value specifies that sub-elements are independent. The order is important and, if possible, they should be shown in the order in which they appear.

alternative: This value specifies that all sub-elements represent alternative versions of the same information. Systems should recognize that the content of the various sub-elements are interchangeable. Systems should choose the ”best” type based on local environment and references, in some cases even through user interaction. The first element is considered to be default or preferred, and the order of remaining elements is not significant. All elements including the first could have a “quality” attribute which could be used to decide which element to show (see section 3.4). In general, it is recommended that preferred element be used (if it is in format client software understands) even if there are elements with a higher “quality” factor. It is assumed that the feed author has choosen it as as reasonable compromise between quality and required bandwidth.

Additional user-defined values of the “kind” attribute are allowed, but their name should start with an “x-” prefix. For example, “x-random”. When an RSS reader encounters a user-defined enclosure “kind” which it does not know how to handle, it should treat the enclosure as “mixed”.

If “kind” was not specified, “mixed” is implied.

3.4 Quality Values

Alternative enclosures (ones inside the “Enclosures” element with “kind” equals to “alternative”) use short “floating point” numbers to indicate their relative importance (“weight”). A weight is a normalized real number in the range 0 through 1, where 0 is the minimum value and 1 the maximum.

Composing agents must not generate more than three digits after the decimal point. User configuration of these values should also be limited in this fashion.

3.5 Transport Values

Each enclosure has a URL of data represented by this enclosure. Usually this information is sufficient to download it. In most cases URL schema (e.g “http:” or “ftp:”) detrmines download protocol.

However some file sharing technologies (most notably "BitTorrent") use special "seed" file which should be downloaded first. This "seed" file contains additional infromation on how real content should be downloaded. This file will be specified by a regular "http:" or "ftp:" URL. However, it is important for a client application processing enclosures to determine if *both* enclsoure content type and transport protocol are supported. In particular, in the case of multiple alternative enclosures it could be used in the decision process to select which one of them will be downloaded.

The goal of the transport attribute is to provide information about the transport type used to download this enclosure. The value of this attribute would be the content type of a "seed" file (e.g. "application/x-bittorrent" for BitTorrent). It should not be confused with the "type" attribute, which would specify content type of actual content ultimately downloaded via this protocol (e.g. "audio/x-realaudio").

The "transport" attribute is optional, and may be omitted in cases where a download URL points directly to the data, not to "seed" file (e.g. direct "http:", "ftp" downloads or "ed2k:" URLs). If omitted, it defaults to the value of "type" attribute.

3.6 Location Values

This proposal makes provisions for mirroring enclosure data on several servers, which can possibly be located in different geographical regions. This would allow one to distribute traffic among them and optmize download performance by choosing a server located near the client.

The "location" attribute could be used to specify spatial location (a place name or geographic co-ordinates) of a server providing enclosure data. This specification does not impose format of this attribute. However it is recommended to use some well-known format or name from common vocabulary. In RSS 2.0 the value must be a single string. In RSS 1.0 "location" property could belong to any well-known RDF class representing spatial information. Examples of usable formats are:

Format	Compatible RSS versions
WGS 84 Geographic Point URI Space	1.0 and 2.0
Getty Thesaurus of Geographic Names	1.0 and 2.0
"address:" URI Scheme	1.0 and 2.0
An RDF Geo vocabulary	1.0
RDFMap	1.0
DCMI Point Encoding Scheme	1.0 and 2.0

In the future, we hope that some of these (or other) formats will emerge as de-facto standards for specifying spatial information in RSS feeds.

The "location" property is optional. If not specified, the client should chose any listed download location randomly.

4 Syntax Definition

4.1 RSS 2.0

Using the RSS 2.0 extension mechanism, we suggest that RSS 2.0 feed producers use "enc:Enclosure" and "enc:Enclosures" elements as described above by using the following namespace definition:

```
xmlns:enc="http://crocodile.org/ns/rss/2.0/enclosures"
```

These elements should be inserted in the feed under the "item" element and should follow the DTD definition included in appendix A.

New elements will not conflict with the existing RSS 2.0 "enclosure" element, since they will be using a different namespace.

4.2 RSS 1.0

We propose a new RSS 1.0 module: "mod_enclosures". This module will use the following namespace definition:

```
xmlns:enc1="http://crocodile.org/ns/rss/1.0/enclosures"
```

It uses RDF vocabulary formally defined by RDF schema[4] included in appenix B.

To specify item enclosures new "enc1:enclosures" property of "item" node is added, which points to "enc1:Enclosure" or "enc1:Enclosures" node with same semantics as in RSS 2.0, but expressed in accordance with RDF/XML syntax.

In the table below you can find correspondence between RSS 2.0 attributes and RDF properties:

Enclosure Type	RSS 2.0 Attribute	RSS 1.0 RDF Property
Simple, Composite	enc:Content-Id	rdf:about
Simple, Composite	enc:rel	enc1:rel
Simple, Composite	enc:description	enc1:description
Simple	enc:url	enc1:url
Simple	enc:type	enc1:type
Simple	enc:transport	enc1:transport
Simple	enc:location	enc1:location
Simple	enc:length	enc1:length
Simple	enc:quality	enc1:quality
Composite	kind	*see below*

Composite enclosures also have the property "enc1:children" pointing to containers with sub-enclosures.

The RSS 2.0 "kind" attribute is expressed via specialized container types pointed by "enc1:children" property. The following table shows correspondence between "enc:kind" attribute values and RDF containers:

RSS 2.0 “kind”	RSS 1.0 RDFS Class
parallel	enc1:Parallel
mixed	rdf:Seq
alternative	rdf:Alt
user-defined	*user-defined subclass of rdfs:Container*

Example of RSS 1.0 feed with enclosures could be seen at page 11.

4.3 ATOM 0.3

At the time of this writing, the current ATOM format specification[6] does not yet define an extensibility model for the format. Once one is defined, we will be able to discuss the integration of the enclosures proposal into it.

5 Differencies from existing “enclosures”

Proposed enclosures syntax differs from the existing RSS 2.0 enclosures already used by many aggregators.

While we were unable to produce full backward compatibility, we tried to model new syntax as closely as possible to the old syntax so as to ease the migration path for RSS software vendors.

In its simplest form, any existing RSS 2.0 enclosure could be converted to one conforming to this proposal simply by using the appropriate XML namespace as described in section 4.1 and using “Enclosure” element name starting with upper case letter.

New syntax should not conflict with the exiting syntax in RSS 2.0, since it uses its own XML namespace. Thus, a feed could contain both old and new-style enclosure definitions.

6 Examples

The following is an example of a simple enclosure for one MPEG audio file for RSS 2.0 feed:

```
<enc:Enclosure
    enc:url="http://www.crocodile.org/sounds/matrosi.mp3"
    enc:length="1939570"
    enc:type="audio/mpeg" />
```

The following example shows two unrelated audio enclosures. They will be played in order they appear.

```
<enc:Enclosures kind="mixed">
   <enc:Enclosure
```

```
        enc:url="http://www.crocodile.org/sounds/matrosi.mp3"
        enc:length="1939570"
        enc:type="audio/mpeg" />
        enc:Content-ID="foo1@bar.net" />
    <enc:Enclosure
        enc:url="http://www.crocodile.org/sounds/ya_segodnya_smeus_nad_soboi.mp3"
        enc:length="1691400"
        enc:Content-ID="foo2@bar.net"
        enc:type="audio/mpeg" />
</enc:Enclosures>
```

An example of composite enclosure for audio files in several alternate formats (with "MP3" being default and most preferable and "WAV" least preferable). Some of files are offered via BitTorrent protocol for download. One file is distributed from two servers, one in US and one in Ukraine. Please note that most preffered format is "mp3" downloaded via BitTorrent.

```
<enc:Enclosures kind="alternative">
    <enc:Enclosure
        enc:url="http://www.crocodile.org/sounds/matrosi.mp3.torrent"
        enc:transport="application/x-bittorrent"
        enc:length="1939570"
        enc:type="audio/mpeg" />
    <enc:Enclosure
        enc:url="http://www.crocodile.org/sounds/matrosi.mp3"
        enc:length="1939570"
        enc:type="audio/mpeg" />
    <enc:Enclosure
        enc:url="http://www.crocodile.org/sounds/matrosi.wav"
        enc:length="8551650"
        enc:quality="0.8"
        enc:location="San Francisco, California, USA"
        enc:type="audio/x-wav" />
    <enc:Enclosure
        enc:url="http://www.crocodile.org.ua/sounds/matrosi.wav"
        enc:length="8551650"
        enc:quality="0.8"
        enc:location="Kiev, Ukraine"
        enc:type="audio/x-wav" />
    <enc:Enclosure
        enc:url="http://www.crocodile.org/sounds/matrosi.ra"
        enc:length="387116"
        enc:quality="0.9"
        enc:type="audio/x-realaudio" />
</enc:Enclosures>
```

The following example, in addition to providing an audio file in multiple alternative formats, also specifies an image (artist photo) which should be shown simultaneously with audio playback:

```
<enc:Enclosures kind="parallel">
   <enc:Enclosures kind="alternative">
       <enc:Enclosure
          enc:url="http://www.crocodile.org/sounds/matrosi.mp3.torrent"
          enc:transport="application/x-bittorrent"
          enc:length="1939570"
          enc:type="audio/mpeg" />
      <enc:Enclosure
          enc:url="http://www.crocodile.org/sounds/matrosi.mp3"
          enc:length="1939570"
          enc:type="audio/mpeg" />
      <enc:Enclosure
          enc:url="http://www.crocodile.org/sounds/matrosi.wav"
          enc:length="8551650"
          enc:quality="0.8"
          enc:location="San Francisco, California, USA"
          enc:type="audio/x-wav" />
      <enc:Enclosure
          enc:url="http://www.crocodile.org.ua/sounds/matrosi.wav"
          enc:length="8551650"
          enc:quality="0.8"
          enc:location="Kiev, Ukraine"
          enc:type="audio/x-wav" />
      <enc:Enclosure
          enc:url="http://www.crocodile.org/sounds/matrosi.ra"
          enc:length="387116"
          enc:type="audio/x-realaudio" />
   </enc:Enclosures>
   <enc:Enclosure
       enc:url="http://www.crocodile.org/vertinsky/Portrait.jpeg"
       enc:length="21713"
       enc:type="image/jpeg" />
</enc:Enclosures>
```

Example of RSS 1.0 feed with the same enclosure using RDF/XML syntax:

```
<?xml version="1.0"  encoding="iso-8859-1"?>
<!DOCTYPE rdf:RDF [<!ENTITY ex "http://crocodile.org/ns/rss/1.0/enc/ex/">]>
<rdf:RDF xmlns="http://purl.org/rss/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:enc1="http://crocodile.org/ns/rss/1.0/enclosures">

  <channel rdf:about="&ex;news.rdf">
    <title>FOO news</title>
    <link>&ex;news.rdf</link>
    <description>Foo company news.</description>
    <items>
      <rdf:Seq>
        <rdf:li rdf:resource="&ex;#2003-03-05-13:42:00" />
      </rdf:Seq>
    </items>
  </channel>

  <item rdf:about="&ex;#2003-03-05-13:42:00">
    <title>Vertinsky Music</title>
    <enc1:enclosures rdf:resource="&ex;#enclosures_top" />
  </item>
```

```
  <enc1:Enclosures rdf:about="&ex;#enclosures_top">
    <enc1:children>
      <enc1:Parallel>
        <rdf:li rdf:resource="&ex;#Portrait.jpeg" />
        <rdf:li rdf:resource="&ex;#enclosures_music" />
      </enc1:Parallel>
    </enc1:children>
  </enc1:Enclosures>

  <enc1:Enclosures rdf:about="&ex;#enclosures_music">
    <enc1:children>
      <rdf:Alt>
        <rdf:li rdf:resource="&ex;#matrosi.mp3.torrent" />
        <rdf:li rdf:resource="&ex;#matrosi.mp3" />
        <rdf:li rdf:resource="&ex;#matrosi.ra" />
        <rdf:li rdf:resource="&ex;#matrosi.wav-us" />
        <rdf:li rdf:resource="&ex;#matrosi.wav-ua" />
      </rdf:Alt>
    </enc1:children>
  </enc1:Enclosures>

  <enc1:Enclosure
       rdf:about="&ex;#Portrait.jpeg"
       enc1:url="http://www.crocodile.org/vertinsky/Portrait.jpeg"
       enc1:length="21713"
       enc1:type="image/jpeg"/>

  <enc1:Enclosure
       rdf:about="&ex;#matrosi.mp3"
       enc1:url="http://www.crocodile.org/sounds/matrosi.mp3"
       enc1:length="1939570"
       enc1:type="audio/mpeg"/>

  <enc1:Enclosure
       rdf:about="&ex;#matrosi.mp3.torrent"
       enc1:url="http://www.crocodile.org/sounds/matrosi.mp3.torrent"
       enc1:length="1939570"
       enc1:type="audio/mpeg"
       enc1:transport="application/x-bittorrent"/>

  <enc1:Enclosure
       rdf:about="&ex;#matrosi.ra"
       enc1:url="http://www.crocodile.org/sounds/matrosi.ra"
       enc1:length="387116"
       enc1:quality="0.9"
       enc1:type="audio/x-realaudio"/>

   <enc1:Enclosure
       rdf:about="&ex;#matrosi.wav-us"
       enc1:url="http://www.crocodile.org/sounds/matrosi.wav"
       enc1:length="8551650"
       enc1:quality="0.8"
       enc1:type="audio/x-wav">
       <enc1:location>
            <dcterms:Point>
                 <rdfs:label>San Francisco</rdfs:label>
                 <rdf:value>
                        name=San Francisco, CA, USA; east=-122.4167; north=37.7667
                 </rdf:value>
            </dcterms:Point>
       </enc1:location>
   </enc1:Enclosure>

   <enc1:Enclosure
       rdf:about="&ex;#matrosi.wav-ua"
       enc1:url="http://www.crocodile.org.ua/sounds/matrosi.wav"
       enc1:length="8551650"
       enc1:quality="0.8"
       enc1:type="audio/x-wav">
       <enc1:location>
            <dcterms:Point>
                 <rdfs:label>Kiev</rdfs:label>
                 <rdf:value>
                        name=Kiev, Ukraine; east=30.5167; north=50.4333
                 </rdf:value>
            </dcterms:Point>
       </enc1:location>
   </enc1:Enclosure>

</rdf:RDF>
```

7 Specifying additional metainformation

While proposed enclosures syntax allows one to reference rich media content in various formats, it has a very limited means of controlling presentation.

For better presentation control, some additional mechanism should be used in addition to this proposal. For example, Synchronized Multimedia Integration Language[15], which allows authors to write interactive multimedia presentations, describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects, and describe the layout of the presentation on a screen. We suggest attaching an SMIL document as one of the enclosures, referencing other enclosures via their Content-IDs.

Another approach would be to extend enclosure elements introduced in this document with additional properties containing additional metadata.

RSS 2.0 enclosure elements could be extended with additional attributes and sub-elements, provided such attributes and sub-elements are defined in separate XML namespace.

RSS 1.0 enclosure elements could be extended with additional properties, provided such properties are defined in separate XML namespace.

8 Related work

There were several earlier enclosures syntax proposal for RSS 1.0. The ones we are aware of are:

- "mod_enc" RDF schema by Suzan Foster
- "RSS Attachment module" by Reto Bachmann-Gmuer
- "BitTorrent RSS Module" by Dave Winer, Adam Curry
- "Podcast RSS Module" by August Trometer, Ray Slakinski
- Some ideas, published by Ian Davis in "rss-dev" mailing list

9 Further Plans

Following open discussion with interested parties, we will work with the approriate groups to discuss the possible inclusion of this syntax into the next revision or RSS 1.0 (as module), RSS 2.0 and ATOM standards.

10 Contact Information

Vadim Zaliva lord@crocodile.org
Alexander Sova bird@noir.crocodile.org

Latest version of this document in PDF format could be found at:

http://www.crocodile.org/lord/RSSenclosures/RSSenclosures.pdf.
Latest version of this document in HTML format could be found at:
http://www.crocodile.org/lord/RSSenclosures/.
Latest version of RDF schema for RSS 1.0 enclosures could be found at:
http://www.crocodile.org/lord/RSSenclosures/enc_schema.rdf.

11 License

References

[1] "RDF Site Summary (RSS) 1.0"

[2] "RDF Site Summary 1.0 Modules"

[3] "RDF/XML Syntax Specification"

[4] "RDF Vocabulary Description Language 1.0: RDF Schema"

[5] "RSS 2.0 Specification"

[6] "The Atom Syndication Format"

[7] "The Atom Publishing Protocol"

[8] "RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies"

[9] "RFC 2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types"

[10] "RFC 2387 - The MIME Multipart/Related Content-type"

[11] "RFC 2392 - Content-ID and Message-ID Uniform Resource Locators"

[12] "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)"

[13] "Tags for the Identification of Languages"

[14] Dave Winer: "Payloads for RSS"

[15] "Synchronized Multimedia Integration Language (SMIL 2.0)"

[16] Creative Commons Attribution-ShareAlike 2.0 License

[17] "Propositions of Conventions for RDF"

[18] "BitTorrent RSS Module"

[19] August Trometer, Ray Slakinski: "Podcast RSS Module"

[20] The Dublin Core Metadata Initiative

[21] WGS 84 Geographic Point URI Space

[22] Getty Thesaurus of Geographic Names

[23] Sean B. Palmer: "*address:* URI Scheme Proposal"

[24] An RDF Geo vocabulary

[25] RDFMap (RDF Mapping Language)

A XML Document Type Definition

Below is XML DTD fragment for proposed enclosures elements in RSS 2.0:

```
<!ELEMENT enc:Enclosures (enc:Enclosure | enc:Enclosures)+ >
<!ATTLIST enc:Enclosures kind (mixed | alternative | parallel) "mixed" >
<!ATTLIST enc:Enclosures rel         CDATA   #IMPLIED >
<!ATTLIST enc:Enclosures description CDATA   #IMPLIED >
<!ATTLIST enc:Enclosures Content-ID  CDATA   #IMPLIED >

<!ELEMENT enc:Enclosure EMPTY >
<!ATTLIST enc:Enclosure  url         CDATA   #REQUIRED >
<!ATTLIST enc:Enclosure  type        CDATA   #REQUIRED >
<!ATTLIST enc:Enclosure  transport   CDATA   #IMPLIED  >
<!ATTLIST enc:Enclosure  location    CDATA   #IMPLIED  >
<!ATTLIST enc:Enclosures rel         CDATA   #IMPLIED  >
<!ATTLIST enc:Enclosures quality     CDATA   "1"       >
<!ATTLIST enc:Enclosure  length      CDATA   #IMPLIED  >
<!ATTLIST enc:Enclosures description CDATA   #IMPLIED  >
<!ATTLIST enc:Enclosure  Content-ID  CDATA   #IMPLIED  >
```

B RDF Schema for Enclosures

Below is enclosures RDF vocabulary definition using RDF Schema[4]:

```
<?xml version="1.0" encoding='us-ascii'?>
<!DOCTYPE rdf:RDF [
          <!ENTITY xsd  "http://www.w3.org/2001/XMLSchema#">
          <!ENTITY dc   "http://purl.org/dc/elements/1.1/">
          <!ENTITY rdf  "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
          <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
          <!ENTITY rss   "http://purl.org/rss/1.0/">
]>
```

```
<!--
     RDF Schema for RSS 1.0 enclosures proposal. Full text of proposal
     could be found at:

     http://www.crocodile.org/lord/RSSenclosures/RSSenclosures.pdf

     Latest version of this schema file could be downloaded from:

     http://www.crocodile.org/lord/RSSenclosures/enc_schema.rdf

     (C)2004 Vadim Zaliva <lord@crocodie.org>,
             Alexander Sova <bird@noir.crocodile.org>

     This work is licensed under a
     Creative Commons Attribution-ShareAlike 2.0 License:
     http://creativecommons.org/licenses/by-sa/2.0/
  -->

<rdf:RDF
  xmlns:rdf  = "&rdf;"
  xmlns:rdfs = "&rdfs;"
  xmlns:dc   = "&dc;"
  xmlns:rss  = "&rss;"
  xml:base   = "http://crocodile.org/ns/rss/1.0/enclosures">

    <!-- Classes -->

    <!-- This is base abstract class which should not be
         used directly in the documents -->
    <rdfs:Class rdf:about="#BaseEnclosure"
                rdfs:label="Base Enclosure">
        <rdfs:subClassOf rdf:resource="&rdf;Resource"/>
    </rdfs:Class>

    <rdfs:Class rdf:about="#Enclosure"
                rdfs:label="Enclosure">
        <rdfs:subClassOf rdf:resource="#BaseEnclosure"/>
    </rdfs:Class>

    <rdfs:Class rdf:about="#Enclosures"
                rdfs:label="Enclosures">
        <rdfs:subClassOf rdf:resource="#BaseEnclosure"/>
    </rdfs:Class>

    <rdfs:Class rdf:about="#Parallel"
                rdfs:label="Parallel">
        <rdfs:subClassOf rdf:resource="&rdfs;Container"/>
    </rdfs:Class>

    <!-- Data types -->
```

```
<rdfs:Datatype rdf:about="&xsd;nonNegativeInteger"/>
<rdfs:Datatype rdf:about="&xsd;float"/>
<rdfs:Datatype rdf:about="&xsd;anyURI"/>
<rdfs:Datatype rdf:about="&xsd;string"/>

<!-- Properties common for 'enclosure' and 'enclosures' -->

<rdf:Property rdf:about="#rel"
              rdfs:label="rel">
    <rdfs:domain rdf:resource="#BaseEnclosure"/>
    <rdfs:range  rdf:resource="&xsd;string"/>
</rdf:Property>

<rdf:Property rdf:about="#description"
              rdfs:label="Description"
              rdfs:comment="A short text description of the enclosure">
    <rdfs:subPropertyOf rdf:resource="&dc;description"/>
    <rdfs:domain rdf:resource="#BaseEnclosure"/>
</rdf:Property>

<!-- Properties specific to 'Enclosure' -->

<rdf:Property rdf:about="#url"
              rdfs:label="URL">
    <rdfs:domain rdf:resource="#Enclosure"/>
    <rdfs:range  rdf:resource="&xsd;anyURI"/>
</rdf:Property>

<rdf:Property rdf:about="#type"
              rdfs:label="Type">
    <rdfs:subPropertyOf rdf:resource="&dc;format"/>
    <rdfs:domain rdf:resource="#Enclosure"/>
</rdf:Property>

<rdf:Property rdf:about="#location"
              rdfs:label="Location">
    <rdfs:subPropertyOf rdf:resource="&dc;coverage"/>
    <rdfs:domain rdf:resource="#Enclosure"/>
</rdf:Property>

<rdf:Property rdf:about="#transport"
              rdfs:label="Transport">
    <rdfs:domain rdf:resource="#Enclosure"/>
    <rdfs:range  rdf:resource="&xsd;string"/>
</rdf:Property>

<rdf:Property rdf:about="#length"
              rdfs:label="Length">
    <rdfs:domain rdf:resource="#Enclosure"/>
```

```
        <rdfs:range  rdf:resource="&xsd;nonNegativeInteger"/>
    </rdf:Property>

    <rdf:Property rdf:about="#quality"
                  rdfs:label="Quality">
        <rdfs:domain rdf:resource="#Enclosure"/>
        <rdfs:range  rdf:resource="&xsd;float"/>
    </rdf:Property>

    <!-- Properties specific to 'Enclosures' -->

    <rdf:Property rdf:about="#children"
                  rdfs:label="Children">
        <rdfs:domain rdf:resource="#Enclosures"/>
        <rdfs:range  rdf:resource="&rdfs;Container"/>
    </rdf:Property>

    <!-- new RSS:item properties -->

    <rdf:Property rdf:about="#enclosures"
                  rdfs:label="RSS Item Enclosures">
        <rdfs:domain rdf:resource="&rss;item"/>
    </rdf:Property>

</rdf:RDF>
```

C Change Log

- November 5, 2004 - Initial version
- November 17, 2004 - Added RDF schema and RSS 1.0 examples
- November 18, 2004
  - Added references to earlier work
  - RDF schema base class for enclosure elements
  - replaced RDF "enc1:kind" property with specialized container types.
  - Changed case of RDF schema class and property names to follow common conventions[17].
  - For consistency, also using uppercased element names in RSS 2.0.
  - Removed "Literal" encoding for RSS 1.0.
  - Cleaned up and simplified RSS 1.0 example.
  - Introduced "quality" property.
- December 13, 2004
  - Introduced "transport" property definition and examples.
  - Introduced "location" property definition and examples.