Really Simple Syndication Best Practices Profile

*Editor's Note: This profile contains a set of recommendations for Really Simple Syndication, a web syndication format documented in **RSS 2.0 (revision 2.0.10)**. This is version 1.0 of this document, published Oct. 15, 2007. The current version will always be available at **this link**. Public comments are welcomed at **RSS-Public**.*

Table of Contents

- 4.1.1.7 **docs**
- 4.1.1.8 **generator**
- 4.1.1.9 **image**
  - 4.1.1.9.1 **link**
  - 4.1.1.9.2 **title**
  - 4.1.1.9.3 **url**
  - 4.1.1.9.4 **description**
  - 4.1.1.9.5 **height**
  - 4.1.1.9.6 **width**
- 4.1.1.10 **language**
- 4.1.1.11 **lastBuildDate**
- 4.1.1.12 **managingEditor**
- 4.1.1.13 **pubDate**
- 4.1.1.14 **rating**
- 4.1.1.15 **skipDays**
  - 4.1.1.15.1 **day**
- 4.1.1.16 **skipHours**
  - 4.1.1.16.1 **hour**
- 4.1.1.17 **textInput**
  - 4.1.1.17.1 **description**
  - 4.1.1.17.2 **link**
  - 4.1.1.17.3 **name**
  - 4.1.1.17.4 **title**
- 4.1.1.18 **ttl**
- 4.1.1.19 **webMaster**
- 4.1.1.20 **item**
  - 4.1.1.20.1 **author**
  - 4.1.1.20.2 **category**
  - 4.1.1.20.3 **comments**
  - 4.1.1.20.4 **description**
  - 4.1.1.20.5 **enclosure**
  - 4.1.1.20.6 **guid**

Simone Carletti
James Holderness
Jenny Levine
Eric Lunt
Randy Charles Morin
Ryan Parman
Jake Savin
Jason Shellen
Paul Querna

OTHER FORMATS

Atom
RDF Site Summary

LINKS

Find on IceRocket

1. Introduction

Really Simple Syndication (RSS) is an XML-based document format for the syndication of web content so that it can be republished on other sites or downloaded periodically and presented to users. The **RSS 2.0 specification** describes how to create RSS documents.

This profile is a set of recommendations for how to create RSS documents that work best in the wide and diverse audience of client software that supports the format. The definitions of the RSS elements in this profile are provided for convenience and MUST NOT be treated as definitive. Refer to the specification for authorititive guidance on the format.

An RSS document, also called a feed, MUST conform to the **XML 1.0** specification and MAY contain elements and attributes defined in a namespace according to the **Namespaces in XML** specification. RSS elements do not belong to a namespace. All elements in an RSS feed that are not defined in a namespace MUST be described in the specification. None of the restrictions described in the specification apply to elements or attributes defined in a namespace.

A **sample feed** demonstrates all of the elements available in RSS. A **namespace sample** shows the same feed extended by the **Creative Commons** and **TrackBack** namespaces.

RSS documents can be tested for validity in the **RSS Validator**.

2. Conventions

In this document, the key words MAY, MUST, MUST NOT, OPTIONAL, RECOMMENDED, REQUIRED, SHALL, SHALL NOT, SHOULD and SHOULD NOT are to be interpreted as described in **RFC 2119**.

Software designed to retrieve and present RSS documents to users is called an aggregator, newsreader or reader. For clarity, this document uses the term aggregator exclusively.

This profile relies primarily on tests conducted with the aggregators Bloglines, BottomFeeder 4.4, FeedDemon 2.5 (2.5.0.10), Google Reader, Microsoft Internet Explorer 7, Mozilla Firefox 2.0 (2.0.9), My Yahoo, NewsGator Online and Opera 9 (9.22).

The test of date-time values was conducted on the aggregators Blogbridge, Bloglines, BottomFeeder, FeedDemon, FeedReader, FeederReader, GreatNews, Internet Explorer 7, JetBrains Omea, Mozilla Thunderbird, Newsgator Online, NewzCrawler, Pluck, RSSBandit, RSSOwl, Sharpreader and Snarfer.

The tests of character data encoding and enclosures also were conducted on Apple Safari 2.0 (2.0.4). Tests of TTL support also employed CITA RSS Aggregator 2.6, NewzCrawler 1.8.0 and Snarfer 0.8.0.

Some sections of the profile rely on surveys of **channel element** and **item element** usage compiled from 1,933 RSS feeds chosen because they appear in OPML subscription lists published on the web.

3. Data Types

The requirements for RSS element and attribute values are described in the sections devoted to each element, aside from the following general requirements.

3.1 Character Data

Requirements

For all elements defined in the RSS specification that enclose character data, the text SHOULD be interpreted as plain text with the exception of an item's **description** element, which MUST be suitable for presentation as HTML. All of these elements MUST NOT contain child elements.

There's no limit on the length of character data that can be contained in an RSS element.

Recommendations

The specification has lacked clarity regarding whether HTML is permitted in elements other than an item's **description**, leading to wide variance in how aggregators treat character data in other elements. This makes it especially difficult for

a publisher to determine how to encode the characters "&" and "<", which must be encoded in XML.

In elements that contain plain text, the form of encoding that works in the widest number of aggregators is using the hexadecimal character reference & to represent "&" and < to represent "<".

Encoding	Presentation
`<title>AT&#x26;T</title>`	AT&T
`<title>Bill &#x26; Ted's Excellent Adventure</title>`	Bill & Ted's Excellent Adventure
`<title>The &#x26;amp; entity</title>`	The & entity
`<title>I &#x3C;3 Phil Ringnalda</title>`	I <3 Phil Ringnalda
`<title>A &#x3C; B</title>`	A < B
`<title>A&#x3C;B</title>`	A<B
`<title>Nice &#x3C;gorilla&#x3E; what's he weigh?</title>`	Nice <gorilla>, what's he weigh?

This form of encoding was presented successfully for all seven of the preceding examples in Apple Safari, Bloglines, Mozilla Firefox and Internet Explorer, six-of-seven in Opera and five-of-seven in FeedDemon and Google Reader. It failed three or more tests in BottomFeeder, My Yahoo and NewsGator Online.

A publisher SHOULD encode "&" and "<" in plain text using hexadecimal character references. When encoding the ">" character, a publisher SHOULD use the hexadecimal reference >.

3.2 Dates and Times

Requirements

All date-time values MUST conform to the RFC 822 **Date and Time Specification** with the exception that a four-digit

year is permitted in addition to a two-digit year.

```
<pubDate>Mon, 15 Oct 2007 14:10:00 GMT</pubDate>
```

```
<lastBuildDate>Mon, 15 Oct 2007 09:10:00 EST</lastBuildDate>
```

```
<pubDate>Mon, 15 Oct 2007 08:10:00 -0600</pubDate>
```

Recommendations

All date-time values SHOULD use a four-digit year.

Although RFC 822 permits multiple spaces and comments between each component in date-time values, most aggregators fail to interpret them correctly. Publishers SHOULD NOT include comments or more than one space between components.

With the exception of "Z", the military time zones in RFC 822 are specified incorrectly and SHOULD NOT be used.

In a test of 18 aggregators, the only date-time values that worked in all of them took one of three forms:

```
Thu, 04 Oct 2007 23:59:45 +0000
Thu, 04 Oct 2007 23:59:45 -0000
Thu, 04 Oct 2007 23:59:45 GMT
```

Each of these values employs Universal Time. The weekday, month and timezone SHOULD be capitalized as shown and the leading zero in the day of the month MAY be omitted.

Most aggregators successfully handled the U.S. time zones defined in RFC 822 and numeric time zones that represent an exact hour.

3.3 E-mail Addresses

Requirements

Several elements MUST contain an e-mail address, but there's no requirement to follow a specific format for such addresses. Publishers could format addresses according to the RFC 2822 **Address Specification**, the RFC 2368 guidelines for **mailto links**, or some other scheme.

Recommendations

The RECOMMENDED format for e-mail addresses in RSS elements is *username@hostname.tld (Real Name)*, as in the

following example:

```
<managingEditor>luksa@dallas.example.com (Frank Luksa)</managingEditor>
```

3.4 URLs

Requirements

In all link and url elements, the first non-whitespace characters in a URL MUST begin with a scheme defined by the **IANA Registry of URI Schemes** such as "ftp://", "http://", "https://", "mailto:" or "news://". These elements MUST NOT contain relative URLs.

Because an aggregator MAY choose which URI schemes to support, publishers of RSS documents MUST NOT assume that all schemes are available.

An **Internationalized Resource Identifier** (IRI) provides a means to identify Internet resources using non-ASCII characters that can't be present in URLs. All link and url elements MUST be valid URLs, so an IRI that contains non-ASCII characters MUST be converted to a URL using the **procedure** described in RFC 3987.

4. Elements

An RSS document consists of the following elements.

4.1. rss

Requirements

The rss element is the top-level element of an RSS feed. A feed that conforms to the RSS specification MUST contain a *version* attribute with the value "2.0".

```
<rss version="2.0">
```

This element is REQUIRED and MUST contain a channel element. The rss element MUST NOT contain more than one channel.

4.1.1 channel

Requirements

The channel element describes the RSS feed, providing such information as its title and description, and contains items that represent discrete updates to the web content represented by the feed.

This element is REQUIRED and MUST contain three child elements: **description**, **link** and **title**.

The channel MAY contain each of the following OPTIONAL elements: **category**, **cloud**, **copyright**, **docs**, **generator**, **image**, **language**, **lastBuildDate**, **managingEditor**, **pubDate**, **rating**, **skipDays**, **skipHours**, **textInput**, **ttl** and **webMaster**.

The preceding elements MUST NOT be present more than once in a channel, with the exception of category.

The channel also MAY contain zero or more **item** elements. The order of elements within the channel MUST NOT be treated as significant.

Recommendations

All item elements SHOULD appear after all of the other elements in a channel.

4.1.1.1 description

Requirements

The description element holds **character data** that provides a human-readable characterization or summary of the feed (REQUIRED).

```
<description>Current headlines from the Dallas Times-Herald newspaper</description>
```

4.1.1.2 link

Requirements

The link element identifies the **URL** of the web site associated with the feed (REQUIRED).

```
<link>http://dallas.example.com</link>
```

4.1.1.3 title

Requirements

The title element holds **character data** that provides the name of the feed (REQUIRED).

```
<title>Dallas Times-Herald</title>
```

Recommendations

If the feed corresponds directly to a web site, the name SHOULD match the name of the site.

4.1.1.4 category

Requirements

The category element identifies a category or tag to which the feed belongs (OPTIONAL).

```
<category>Media</category>
```

This element MAY include a *domain* attribute that identifies the taxonomy in which the category is placed.

```
<category domain="dmoz">News/Newspapers/Regional/United_States/Texas</category>
```

A channel MAY contain more than one category element.

Recommendations

The category's value SHOULD be a slash-delimited string that identifies a hierarchical position in the taxonomy.

4.1.1.5 cloud

Requirements

The cloud element indicates that updates to the feed can be monitored using a web service that implements the **RssCloud application programming interface** (OPTIONAL).

The element MUST have five attributes that describe the service:

- The *domain* attribute identifies the host name or IP address of the web service that monitors updates to the feed.
- The *path* attribute provides the web service's path.
- The *port* attribute identifies the web service's TCP port.
- The *protocol* attribute MUST contain the value "xml-rpc" if the service employs XML-RPC or "soap" if it employs

SOAP.

- The *registerProcedure* attribute names the remote procedure to call when requesting notification of updates.

```
<cloud domain="server.example.com" path="/rpc" port="80" protocol="xml-rpc"
registerProcedure="cloud.notify" />
```

In this example, an aggregator could request notification by calling the cloud.notify method of the XML-RPC web service at server.example.com, port 80, path /rpc.

This element is an empty element defined by a single tag and its attributes, unless extended by a namespace.

4.1.1.6 copyright

Requirements

The copyright element declares the human-readable copyright statement that applies to the feed (OPTIONAL).

```
<copyright>Copyright 2007 Dallas Times-Herald</copyright>
```

Recommendations

When a feed lacks a copyright element, aggregators SHOULD NOT assume that is in the public domain and can be republished and redistributed without restriction. Under the **Berne Convention** adopted by the United States and more than 150 other countries, a work does not require a copyright statement to be protected by copyright.

4.1.1.7 docs

Requirements

The docs element identifies the **URL** of the RSS specification implemented by the software that created the feed (OPTIONAL).

```
<docs>http://www.rssboard.org/rss-specification</docs>
```

Recommendations

If you are relying on the specification and profile published by the **RSS Advisory Board**, the value of this element SHOULD be "http://www.rssboard.org/rss-specification".

4.1.1.8 generator

Requirements

The generator element credits the software that created the feed (OPTIONAL).

```
<generator>Microsoft Spaces v1.1</generator>
```

4.1.1.9 image

Requirements

The image element supplies a graphical logo for the feed (OPTIONAL).

The image MUST contain three child elements: **link**, **title** and **url**. It also MAY contain three OPTIONAL elements: **description**, **height** and **width**.

```
<image>
  <link>http://dallas.example.com</link>
  <title>Dallas Times-Herald</title>
  <url>http://dallas.example.com/masthead.gif</url>
  <description>Read the Dallas Times-Herald</description>
  <height>32</height>
  <width>96</width>
</image>
```

4.1.1.9.1 link

Requirements

The image's link element identifies the **URL** of the web site represented by the image (REQUIRED).

Recommendations

This SHOULD be the same URL as the channel's **link** element.

4.1.1.9.2 title

Requirements

The image's title element holds **character data** that provides a human-readable description of the image (REQUIRED).

Recommendations

This element SHOULD have the same text as the channel's **title** element and be suitable for use as the **alt** attribute of the **img** tag in an HTML rendering.

4.1.1.9.3 url

Requirements

The image's url element identifies the **URL** of the image, which MUST be in the GIF, JPEG or PNG formats (REQUIRED).

4.1.1.9.4 description

Requirements

The image's description element holds **character data** that provides a human-readable characterization of the site linked to the image (OPTIONAL).

Recommendations

The description SHOULD be suitable for use as the **title** attribute of the **a** tag in an HTML rendering.

4.1.1.9.5 height

Requirements

The image's height element contains the height, in pixels, of the image (OPTIONAL). The image MUST be no taller than 400 pixels. If this element is omitted, the image is assumed to be 31 pixels tall.

4.1.1.9.6 width

Requirements

The image's width element contains the width, in pixels, of the image (OPTIONAL). The image MUST be no wider than 144 pixels. If this element is omitted, the image is assumed to be 88 pixels wide.

4.1.1.10 language

Requirements

The channel's language element identifies the natural language employed in the feed (OPTIONAL).

The language MUST be identified using one of the **RSS language codes** or a **language code** permitted by the World Wide Web Consortium for use in HTML. The U.S. Library of Congress publishes the current list of **ISO 639 language codes** adopted by HTML.

```
<language>epo</language>
```

4.1.1.11 lastBuildDate

Requirements

The channel's lastBuildDate element indicates the last **date and time** the content of the feed was updated (OPTIONAL).

```
<lastBuildDate>Sun, 14 Oct 2007 17:17:44 GMT</lastBuildDate>
```

4.1.1.12 managingEditor

Requirements

The channel's managingEditor element provides the **e-mail address** of the person to contact regarding the editorial content of the feed (OPTIONAL).

```
<managingEditor>jlehrer@dallas.example.com (Jim Lehrer)</managingEditor>
```

4.1.1.13 pubDate

Requirements

The channel's pubDate element indicates the publication **date and time** of the feed's content (OPTIONAL). Publishers of daily, weekly or monthly periodicals could use this element to associate feed items with the date they most recently went to press.

```
<pubDate>Sun, 14 Oct 2007 05:00:00 GMT</pubDate>
```

4.1.1.14 rating

Requirements

The channel's rating element supplies an advisory label for the content in a feed, formatted according to the specification for the **Platform for Internet Content Selection (PICS)** (OPTIONAL).

```
<rating>(PICS-1.1 "http://www.rsac.org/ratingsv01.html" l by "webmaster@example.com" on
"2007.01.29T10:09-0800" r (n 0 s 0 v 0 l 0))</rating>
```

Recommendations

This element gets little usage among publishers, appearing in fewer than one percent of surveyed feeds.

4.1.1.15 skipDays

Requirements

The channel's skipDays element identifies days of the week during which the feed is not updated (OPTIONAL). This element contains up to seven day elements identifying the days to skip.

Recommendations

Although an aggregator SHOULD NOT request the feed on the days identified by this element, the point is largely moot because of how infrequently it is used by publishers. Fewer than one percent of surveyed feeds included a skipDays element.

4.1.1.15.1 day

Requirements

The day element identifies a weekday in Greenwich Mean Time (GMT) (REQUIRED). Seven values are permitted -- "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" or "Sunday" -- and MUST NOT be duplicated.

```
<skipDays>
  <day>Saturday</day>
  <day>Sunday</day>
</skipDays>
```

4.1.1.16 skipHours

Requirements

The channel's skipHours element identifies the hours of the day during which the feed is not updated (OPTIONAL). This element contains individual hour elements identifying the hours to skip.

Recommendations

An aggregator SHOULD NOT request the feed on the hours identified by this element.

4.1.1.16.1 hour

Requirements

The hour element identifies an hour of the day in Greenwich Mean Time (GMT) (REQUIRED). The hour MUST be expressed as an integer representing the number of hours since 00:00:00 GMT. Values from 0 to 23 are permitted, with 0 representing midnight. An hour MUST NOT be duplicated.

```
<skipHours>
  <hour>0</hour>
  <hour>1</hour>
  <hour>2</hour>
  <hour>22</hour>
  <hour>23</hour>
</skipHours>
```

Recommendations

RSS specifications differ in the number assigned to midnight, which is 0 in the current RSS specification and 24 in **RSS 0.91**. For this reason, aggregators SHOULD accept both 0 and 24 to represent midnight.

4.1.1.17 textInput

Requirements

The textInput element defines a form to submit a text query to the feed's publisher over the Common Gateway Interface (CGI) (OPTIONAL).

The element MUST contain a **description**, **link**, **name** and **title** child element.

```
<textInput>
  <description>Your aggregator supports the textInput element. What software are you using?
</description>
  <link>http://www.cadenhead.org/textinput.php</link>
  <name>query</name>
  <title>TextInput Inquiry</title>
</textInput>
```

Recommendations

The RSS specification actively discourages publishers from using the textInput element, calling its purpose "something of a mystery" and stating that "most aggregators ignore it." Fewer than one percent of surveyed RSS feeds included the element. The only aggregators known to support it are BottomFeeder and Liferea.

For this reason, publishers SHOULD NOT expect it to be supported in most aggregators.

4.1.1.17.1 description

Requirements

The input form's description element holds **character data** that provides a human-readable label explaining the form's purpose (REQUIRED).

4.1.1.17.2 link

Requirements

The input form's link element identifies the **URL** of the CGI script that handles the query (REQUIRED).

4.1.1.17.3 name

Requirements

The input form's name element provides the name of the form component that contains the query (REQUIRED). The name MUST begin with a letter and contain only these characters: the letters A to Z in either case, numeric digits, colons (":"), hyphens ("-"), periods (".") and underscores ("_").

4.1.1.17.4 title

Requirements

The input form's title element labels the button used to submit the query (REQUIRED).

4.1.1.18 ttl

Requirements

The channel's ttl element represents the feed's time to live (TTL): the maximum number of minutes to cache the data before an aggregator requests it again (OPTIONAL).

```
<ttl>60</ttl>
```

Recommendations

By convention, most aggregators check an RSS feed for updates once an hour. The ttl, **skipDays** and **skipHours** elements provide a means for publishers to offer guidance regarding a feed's frequency of updates.

Twenty-one percent of surveyed feeds included a ttl element. Support for the element appears sparse among aggregators, perhaps due to disagreement over its meaning in the RSS specification.

The following aggregators ignore this element: BlogBridge, Bloglines, Google Reader, JetBrains Omea, Mozilla Firefox, My Yahoo, NewsGator Online and RSSBandit.

Most aggregators that support TTL use its value as the maximum frequency of update checks. Seven aggregators won't check a feed more frequently than its TTL: BottomFeeder, CITA RSS Aggregator, GreatNews, Internet Explorer 7, NewzCrawler, Opera 9 and Snarfer.

One aggregator, FeedDemon, employs the TTL value as the recommended frequency of checks, as long as it's 30 minutes or higher and has not been overridden by a user.

No aggregators have been found that use the TTL as the minimum frequency of checks, as intended by the specification.

Because of these differences, aggregators that support this element SHOULD treat it as a publisher's suggestion of a feed's update frequency, not a hard rule. For instance, an aggregator that gives users the ability to choose how often to check a feed could use its TTL as the default value.

4.1.1.19 webMaster

Requirements

The channel's webMaster element provides the **e-mail address** of the person to contact about technical issues regarding the feed (OPTIONAL).

```
<webMaster>helpdesk@dallas.example.com</webMaster>
```

4.1.1.20 item

Requirements

An item element represents distinct content published in the feed such as a news article, weblog entry or some other form of discrete update. A channel MAY contain any number of items (or no items at all).

An item MAY contain the following child elements: **author**, **category**, **comments**, **description**, **enclosure**, **guid**, **link**, **pubDate**, **source** and **title**. All of these elements are OPTIONAL but an item MUST contain either a title or description.

The preceding elements MUST NOT be present more than once in an item, with the exception of category.

```
<item>
  <title>Seventh Heaven! Ryan Hurls Another No Hitter</title>
  <link>http://dallas.example.com/1991/05/02/nolan.htm</link>
  <description>Texas Rangers pitcher Nolan Ryan hurled the seventh no-hitter of his
legendary career on Arlington Appreciation Night, defeating the Toronto Blue Jays 3-0. The
44-year-old struck out 16 batters before a crowd of 33,439.</description>
</item>
```

4.1.1.20.1 author

Requirements

An item's author element provides the **e-mail address** of the person who wrote the item (OPTIONAL).

```
<author>jbb@dallas.example.com (Joe Bob Briggs)</author>
```

Recommendations

A feed published by an individual SHOULD omit this element and use the **managingEditor** or **webMaster** channel elements to provide contact information.

4.1.1.20.2 category

Requirements

An item's category element identifies a category or tag to which the item belongs (OPTIONAL).

```
<category>movies</category>
```

This element MAY include a *domain* attribute that identifies the category's taxonomy.

```
<category domain="rec.arts.movies.reviews">1983/V</category>
```

An item MAY contain more than one category element.

Recommendations

The category's value SHOULD be a slash-delimited string that identifies a hierarchical position in the taxonomy.

4.1.1.20.3 comments

Requirements

An item's comments element identifies the **URL** of a web page that contains comments received in response to the item (OPTIONAL).

```
<comments>http://dallas.example.com/feedback/1983/06/joebob.htm</comments>
```

4.1.1.20.4 description

Requirements

An item's description element holds **character data** that contains the item's full content or a summary of its contents, a decision entirely at the discretion of the publisher. This element is OPTIONAL if the item contains a **title** element.

```
<description>I'm headed for France. I wasn't gonna go this year, but then last week "Valley
Girl" came out and I said to myself, Joe Bob, you gotta get out of the country for a while.
</description>
```

The description MUST be suitable for presentation as HTML. HTML markup MUST be encoded as character data either by employing the HTML entities < ("<") and > (">") or a **CDATA** section.

Escaped markup created with character entities:

```
<description>I'm headed for France. I wasn't gonna go this year, but then last week &lt;a
href="http://www.imdb.com/title/tt0086525/"&gt;Valley Girl&lt;/a&gt; came out and I said to
myself, Joe Bob, you gotta get out of the country for a while.</description>
```

CDATA encoding:

```
<description><![CDATA[I'm headed for France. I wasn't gonna go this year, but then last
week <a href="http://www.imdb.com/title/tt0086525/">Valley Girl</a> came out and I said to
myself, Joe Bob, you gotta get out of the country for a while.]]></description>
```

Recommendations

The description SHOULD NOT contain relative URLs, because the RSS format does not provide a means to identify the base URL of a document. When a relative URL is present, an aggregator MAY attempt to resolve it to a full URL using the channel's **link** as the base.

4.1.1.20.5 enclosure

Requirements

An item's enclosure element associates a media object such as an audio or video file with the item (OPTIONAL). The element MUST have three attributes:

- The *length* attribute indicates the size of the file in bytes
- The *type* attribute identifies the file's **MIME media type**
- The *url* attribute identifies the **URL** of the file

```
<enclosure length="24986239" type="audio/mpeg"
url="http://dallas.example.com/joebob_050689.mp3" />
```

The enclosure element is an empty element defined by a single tag and its attributes, unless extended by a namespace.

Recommendations

Support for the enclosure element in RSS software varies significantly because of disagreement over whether the specification permits more than one enclosure per item. Although the author intended to permit no more than one enclosure in each item, this limit is not explicit in the specification.

Blogware, Movable Type and WordPress enable publishers to include multiple enclosures in each item of their RSS documents. This works successfully in some aggregators, including BottomFeeder, FeederReader, NewsGator and

Safari.

Other software does not support multiple enclosures, including Bloglines, FeedDemon, Google Reader and Microsoft Internet Explorer 7. The first enclosure is downloaded automatically, an aspect of enclosure support relied on in podcasting, and the additional enclosures are either ignored or must be requested manually.

For best support in the widest number of aggregators, an item SHOULD NOT contain more than one enclosure.

Though an enclosure MUST specify its size with the *length* attribute, the size of some media objects cannot be determined by an RSS publisher. Examples include the streaming media formats RealAudio and Apple QuickTime.

When an enclosure's size cannot be determined, a publisher SHOULD use a length of 0.

The peer-to-peer file-sharing protocol BitTorrent deploys files using a small key file called a torrent that tells a client how to find and download the file.

When an enclosure is delivered in a multi-step process like the one used by BitTorrent, the length SHOULD be the size of the first file that must be downloaded to begin the process.

4.1.1.20.6 guid

Requirements

An item's guid element provides a string that uniquely identifies the item (OPTIONAL). The guid MAY include an *isPermaLink* attribute.

The guid enables an aggregator to detect when an item has been received previously and does not need to be presented to a user again. If the guid's *isPermaLink* attribute is omitted or has the value "true", the guid MUST be the permanent **URL** of the web page associated with the item.

```
<guid>http://dallas.example.com/1983/05/06/joebob.htm</guid>
```

If the guid's *isPermaLink* attribute has the value "false", the guid MAY employ any syntax the feed's publisher has devised for ensuring the uniqueness of the string, such as the **Tag URI scheme** described in RFC 4151.

```
<guid isPermaLink="false">tag:dallas.example.com,4131:news</guid>
```

Requirements

A publisher SHOULD provide a guid with each item.

4.1.1.20.7 link

Requirements

An item's link element identifies the **URL** of a web page associated with the item (OPTIONAL).

```
<link>http://dallas.example.com/1983/05/06/joebob.htm</link>
```

4.1.1.20.8 pubDate

Requirements

An item's pubDate element indicates the publication **date and time** of the item (OPTIONAL).

```
<pubDate>Fri, 05 Oct 2007 09:00:00 CST</pubDate>
```

Recommendations

The specification recommends that aggregators SHOULD ignore items with a publication date that occurs in the future, providing a means for publishers to embargo an item until that date.

None of the tested aggregators withheld an item with a future publication date from readers. For this reason, publishers SHOULD NOT include items in a feed until they are ready for publication.

4.1.1.20.9 source

Requirements

An item's source element indicates the fact that the item has been republished from another RSS feed (OPTIONAL). The element MUST have a *url* attribute that identifies the **URL** of the source feed.

The value of the source is the **title** of the source feed.

```
<source url="http://la.example.com/rss.xml">Los Angeles Herald-Examiner</source>
```

4.1.1.20.10 title

Requirements

An item's title element holds **character data** that provides the item's headline. This element is OPTIONAL if the item contains a **description** element.

```
<title>Joe Bob Goes to the Drive-In</title>
```

5. Namespace Elements

The RSS specification encourages the extension of the format through the use of namespaces. Some namespace elements serve a similar purpose to RSS elements defined in the specification, which raises the question of how aggregators should treat a feed in which both are present.

This section of the profile contains recommendations for how to handle these situations. This MUST NOT be considered definitive in regard to namespace elements, which are defined by their authors.

5.1. Atom

The **Atom** syndication format, which serves a similar purpose to RSS, offers some elements closely comparable to RSS elements and others that provide new capabilities. Any of these elements can be used in RSS by employing Atom as a namespace.

This namespace requires the "http://www.w3.org/2005/Atom" declaration in the **rss** element.

```
<rss xmlns:atom="http://www.w3.org/2005/Atom">
```

5.1.1 atom:link

Requirements

The atom:link element defines a relationship between a web resource (such as a page) and an RSS **channel** or **item** (OPTIONAL). The most common use is to identify an HTML representation of an entry in an RSS or Atom feed.

The element MUST have an *href* attribute that contains the **URL** of the related resource and MAY contain the following attributes:

- The *hreflang* attribute identifies the language used by the related resource using an HTML **language code**
- The *length* attribute contains the resource's size, in bytes
- The *title* attribute provides a human-readable description of the resource
- The *type* attribute identifies the resource's **MIME media type**

The element also MAY contain a *rel* attribute, which contains a keyword that identifies the nature of the relationship between the linked resouce and the element. Five relationships are possible:

- The value "alternate" describes an alternate representation, such as a web page containing the same content as a feed entry
- The value "enclosure" describes a a media object such as an audio or video file
- The value "related" describes a related resource
- The value "self" describes the feed itself
- The value "via" describes the original source that authored the entry, when it's not the feed publisher

Recommendations

An RSS feed can identify its own **URL** using the atom:link element within a **channel**. The link MUST have the *rel* attribute "self", an *href* attribute containing the feed's URL and MAY have a *type* attribute of "application/rss+xml":

```
<atom:link href="http://dallas.example.com/rss.xml" rel="self" type="application/rss+xml"
/>
```

There's no means to do this with RSS elements defined in the specification. Identifying a feed's URL within the feed makes it more portable, self-contained, and easier to cache. For these reasons, a feed SHOULD contain an atom:link used for this purpose.

The other uses of atom:link are closely analogous to the channel **link** element and the item **enclosure**, **link** and **source** elements.

When a namespace element duplicates the functionality of an element defined in RSS, the core element SHOULD be used.

5.2. Content

The **Content** namespace offers a means of defining item content with more precision than the **description** element.

This namespace requires the "http://purl.org/rss/1.0/modules/content/" declaration in the **rss** element.

```
<rss xmlns:content="http://purl.org/rss/1.0/modules/content/">
```

5.2.1 content:encoded

Requirements

The content:encoded element defines the full content of an **item** (OPTIONAL). This element has a more precise purpose than the **description** element, which can be the full content, a summary or some other form of excerpt at the publisher's discretion.

The content MUST be suitable for presentation as HTML and be encoded as character data in the same manner as the **description** element.

Recommendations

The content:encoded element can be used in conjunction with the description element to provide an item's full content along with a shorter summary. Under this approach, the complete text of the item is presented in content:encoded and the summary in description.

All of the tested aggregators support the use of content:encoded to define an item's content with the exception of Mozilla Firefox 2.0, which only displays items by their title.

Only one of the aggregators, Bloglines, supports the related use of description as a summary.

When an **item** contains both elements, six of the eight aggregators display content:encoded and ignore description. FeedDemon displays the element defined last within the item element and Firefox 2.0 doesn't support descriptions.

Publishers who don't want to employ item summaries in their feeds SHOULD use the description element for an item's full content rather than content:encoded because it has the widest support.

Publishers who employ summaries SHOULD store the summary in description and the full content in content:encoded, ordering description first within the item. On items with no summary, the full content SHOULD be stored in description.

5.3. Dublin Core

The **Dublin Core** namespace supports the **Dublin Core Metadata Initiative**, a standard for describing resources on the Internet to identify their author, date of publication, publisher and similar information.

This namespace requires the "http://purl.org/dc/elements/1.1/" declaration in the **rss** element.

```
<rss xmlns:dc="http://purl.org/dc/elements/1.1/">
```

5.3.1 dc:creator

Requirements

The dc:creator element identifies the person or entity who wrote an **item** (OPTIONAL). An item MAY contain more than

one dc:creator element to credit multiple authors.

The creator can be identified using a real name, username or some other means of identification at the publisher's discretion.

```
<dc:creator>Joe Bob Briggs</dc:creator>
```

Recommendations

The value of the dc:creator element is less restrictive than the **author** element, which must contain an e-mail address. Publishers often rely on dc:creator to credit authorship without revealing e-mail addresses in a form that can be exploited by spammers.

All of the tested aggregators that display item authors support both the author and dc:creator elements. (BottomFeeder, Mozilla Firefox 2.0 and My Yahoo do not identify authors.)

When an **item** contains both elements, aggregators handle it in different ways. Some take the first element that appears within the item, others take the last and one aggregator combines their values.

Publishers SHOULD use author when they want to reveal an author's e-mail address and dc:creator when they don't. The same item SHOULD NOT include both elements.

This same recommendation SHOULD be followed for the use of dc:creator with the channel elements **managingEditor** and **webMaster**.

5.4. Slash

The **Slash** namespace supports features associated with items in the **Slash** content-management system, the software that powers **Slashdot** and other sites. Any RSS-producing software that employs the same features can use the namespace.

This namespace requires the "http://purl.org/rss/1.0/modules/slash/" declaration in the **rss** element.

```
<rss xmlns:slash="http://purl.org/rss/1.0/modules/slash/">
```

5.4.1 slash:comments

Requirements

The slash:comments element contains a non-negative integer that counts the number of comments that an **item** has

received (OPTIONAL).

```
<slash:comments>13</slash:comments>
```

This complements the **comments** element, which identifies the **URL** of the web page where an item's comments are displayed.

Recommendations

On an active web site, comment counts change frequently as new comments are published, so by necessity this element contains a snapshot of the totals at a particular moment in time. Because the Slash namespace lacks an element to indicate when the comment counts were compiled, publishers who use this element also SHOULD include a **lastBuildDate** element.

6. License

7. Credits

The author of this document is the **RSS Advisory Board**. Comments and corrections regarding this document are encouraged on the **RSS-Public** mailing list.

The RSS format was created by Dan Libby and Ramanathan V. Guha. The Atom syndication format was created by the AtomPub Working Group. The Content namespace was created by Gabe Beged-Dov, Aaron Swartz and Eric van der Vlist. The Dublin Core namespace was created by Beged-Dov, Dan Brickley, Rael Dornfest, Ian Davis, Leigh Dodds, Jonathan Eisenzopf, David Galbraith, R.V. Guha, Ken MacLeod, Eric Miller, Swartz and van der Vlist. The Slash namespace was created by Chris Nandor and Dornfest.

Rogers Cadenhead, James Holderness, Randy Charles Morin and Geoffrey Sneddon contributed to this document.