RDF Site Summary (RSS) 1.0

Abstract

RDF Site Summary (RSS) is a lightweight multipurpose extensible metadata description and syndication format. RSS is an XML application, conforms to the W3C's RDF Specification and is extensible via XML-namespace and/or RDF based modularization.

Authors

The members of the RSS-DEV Working Group:

Gabe Beged-Dov, JFinity Systems LLC
Dan Brickley, ILRT
Rael Dornfest, O'Reilly & Associates
Ian Davis, Calaba, Ltd.
Leigh Dodds, xmlhack
Jonathan Eisenzopf, Whirlwind Interactive
David Galbraith, Moreover.com
R.V. Guha, guha.com
Ken MacLeod, (Independent)
Eric Miller, Online Computer Library Center, Inc.
Aaron Swartz, The Info Network
Eric van der Vlist, Dyomedea

Version

Latest Version: http://purl.org/rss/1.0/spec

1.3.4 2001-05-30 Fixed small typo in section 5.3.6 (as, announcement)
1.3.3 2001-03-20 Updated mime-type and URI (as, announcement)
1.3.2 2000-12-19 (Changed style and tidied markup; revisions author: SBP)
1.3.1 2000-12-17 (Typo correction: An upper limit of 15 items per RSS document is recommended, not enforced [5.5].)
1.3 2000-12-09

Status

Release

Comments should be directed to the RSS-DEV mailing list, archived at http://www.egroups.com/messages/rss-dev.

Rights

Table of Contents

1. Introduction

RDF Site Summary (RSS) is a lightweight multipurpose extensible metadata description and syndication format. RSS is an XML application, conforming to the W3C's RDF Specification. RSS is extensible via XML-namespace and/or RDF based modularization.

An RSS summary, at a minimum, is a document describing a "channel" consisting of URL-retrievable items. Each item consists of a title, link, and brief description. While items have traditionally been news headlines, RSS has seen much

repurposing in its short existence. For sample RSS 1.0 documents, see the Examples section below.

2. Background

RSS 0.9 was introduced in 1999 by Netscape as a channel description framework / content-gathering mechanism for their My Netscape Network (MNN) portal. By providing a simple snapshot-in-a-document, web site producers acquired audience through the presence of their content on My Netscape.

A by-product of MNN's work was RSS's use as an XML-based lightweight syndication format, quickly becoming a viable alternative to ad hoc syndication systems and practical in many scenarios where heavyweight standards like ICE were overkill. And the repurposing didn't stop at headline syndication; today's RSS feeds carry an array of content types: news headlines, discussion forums, software announcements, and various bits of proprietary data.

RSS 0.91, re-dubbed "Rich Site Summary," followed shortly on the heels of 0.9. It had dropped its roots in RDF and sported new elements from Userland's scriptingNews format -- most notably being a new item-level <description> element, bringing RSS into the (lightweight) content syndication arena.

While Netscape discontinued its RSS efforts, evangelism by Userland's Dave Winer led to a groundswell of RSS-as-syndication-framework adoption. Inclusion of RSS 0.91 as one of the syndicaton formats for its Manila product and related EditThisPage.com service brought together the weblog and syndication worlds.

3. Motivation

As RSS continues to be re-purposed, aggregated, and categorized, the need for an enhanced metadata framework grows. Channel- and item-level title and description elements are being overloaded with metadata and HTML. Some producers are even resorting to inserting unofficial ad hoc elements (e.g., <category>, <date>, <author>) in an attempt to augment the sparse metadata facilities of RSS.

One proposed solution is the addition of more simple elements to the RSS core. This direction, while possibly being the simplest in the short run, sacrifices scalability and requires iterative modifications to the core format, adding requested and removing unused functionality. See Ian Davis's RSS Survey (2000-07-25) for a more concrete representation of element usage.

A second solution, and the one adopted here, is the compartmentalization of specific functionality into the pluggable RSS modules. This is one of the approaches used in this specification: modularization is achieved by using XML Namespaces for partitioning vocabularies. Adding and removing RSS functionality is then just a matter of the inclusion of a particular set of modules best suited to the task at hand. No reworking of the RSS core is necessary.

Advanced applications of RSS are demanding richer respresentation of relationships between intra- and inter-channel elements (e.g. threaded discussions). RDF (Resource Description Framework) provides a framework for just such rich metadata modeling. RSS 0.9 provided a basic (albeit limited) RDF base upon which to layer further structure.

4. Design Goals

The RSS 1.0 design goal is an XML-based lightweight multipurpose extensible metadata description and syndication format. Backward compatibility with RSS 0.9 is a goal for ease of adoption by existing syndicated content producers.

4.1 Lightweight

Much of RSS's success stems from the fact that it is simply an XML document rather than a full syndication framework such as XMLNews and ICE.

The following is a basic sample RSS 1.0 document, making use of only the core RSS 1.0 element set.

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/"
>

  <channel rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>
      XML.com features a rich mix of information and services
      for the XML community.
    </description>

    <image rdf:resource="http://xml.com/universal/images/xml_tiny.gif" />

    <items>
      <rdf:Seq>
        <rdf:li resource="http://xml.com/pub/2000/08/09/xslt/xslt.html" />
        <rdf:li resource="http://xml.com/pub/2000/08/09/rdfdb/index.html" />
      </rdf:Seq>
    </items>

  </channel>

  <image rdf:about="http://xml.com/universal/images/xml_tiny.gif">
    <title>XML.com</title>
    <link>http://www.xml.com</link>
    <url>http://xml.com/universal/images/xml_tiny.gif</url>
  </image>

  <item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
    <title>Processing Inclusions with XSLT</title>
    <link>http://xml.com/pub/2000/08/09/xslt/xslt.html</link>
    <description>
     Processing document inclusions with general XML tools can be
     problematic. This article proposes a way of preserving inclusion
     information through SAX-based processing.
    </description>
  </item>

  <item rdf:about="http://xml.com/pub/2000/08/09/rdfdb/index.html">
    <title>Putting RDF to Work</title>
    <link>http://xml.com/pub/2000/08/09/rdfdb/index.html</link>
    <description>
     Tool and API support for the Resource Description Framework
     is slowly coming of age. Edd Dumbill takes a look at RDFDB,
     one of the most exciting new RDF toolkits.
    </description>
  </item>

</rdf:RDF>
```

4.2 Multipurpose

The 12 months since version 0.91 was released have seen the surfacing of various novel uses for RSS. RSS is being called upon to evolve with growing application needs: aggregation, discussion threads, job listings, homes for sale (multiple listings services), sports scores, document cataloging, etc. Via XML-namespace based modularization and RDF, RSS 1.0 builds a framework for both standardized and ad hoc re-purposing.

4.3 Extensible

The crux of the difference between RSS 1.0 and earlier (or lateral) versions lies in its extensibility via XML Namespaces and RDF (Resource Description Framework) compliance.

Namespace-based modules allow compartmentalized extensibility. This allows RSS to be extended:

- without need of iterative rewrites of the core specification
- without need of consensus on each and every element
- without bloating RSS with elements the majority of which won't be used in any particular arena or application
- without naming collisions

RSS modules are covered in more detail in the modules section below.

4.4 Metadata

Metadata is data about data. While there is no dearth of information floating about the Web, there is precious little description thereof. The W3C's Metadata Activity Statement has this to say on the subject:

> The possible uses of the Web seem endless, but there the technology is missing a crucial piece. Missing is a part of the Web which contains information about information - labeling, cataloging and descriptive information structured in such a way that allows Web pages to be properly searched and processed in particular by computer.

RDF allows for representation of rich metadata relationships beyond what is possible with earlier flat-structured RSS. The existing RDF base in RSS 0.9 was the reason for choosing to build on the earlier version of RSS; attempting to re-introduce RDF into RSS version 0.91 proved a "putting the toothpaste back into the tube" proposition.

4.5 Syndication

Syndication is here defined as making data available online for retrieval and further transmission, aggregation, or online publication. The specifics of the various intricacies of syndication systems (free vs. subscription, push vs. pull, etc.) is beyond the scope of this specification.

5. Core Syntax

The core of RSS 1.0 is built upon RSS 0.9. RSS 1.0's focus is on extensibility through XML-namespaces and RDF whilst maintaining backward compatibility.

Backward Compatibility with RSS 0.9
Backward compatibility is accomplished by the assumption and stipulation that basic RSS parsers, modules, and libraries ignore what they weren't designed to understand:

1. Attributes; RSS 0.9 has no attributes outside of the RDF namespace declarations.
2. Element members of modularized extensions residing outside the default namespace.
3. Ad-hoc elements that don't interfere with the overall structure of the RSS 0.9 document.

Extensibility via XML Namespace-Based Modularization

RSS 1.0 is extensible through XML-namespace based modules. While ad hoc extensibility is of course encouraged, it is hoped that a core set of agreed-upon modules covering such functionality as taxonomy, aggregation, Dublin Core, etc will emerge. See the Modules section below, as well as the registry of core RSS 1.0 Modules.

One restriction imposed on sub-elements of top-level channel, image, item, and textinput elements [5.3 <channel>, 5.4 <image>, 5.5 <item>, 5.6 <textinput>] is that these elements may not contain repeating sub-elements (e.g. `<item><dc:subject /><dc:subject /></item>`). This proposal only constrains the immediate sub-elements. Any further depth (of rich content or repeated elements) is already well-defined using RDF syntax.

RDF
RSS 1.0 builds on the fledgling RDF framework found in RSS 0.9 (and lost in RSS 0.91) via the following minimal additions:

- Each second-level element (channel, image, item, and textinput) must include an *`rdf:about`* attribute 5.3, 5.4, 5.5, 5.6].
- A channel-level RDF table of contents associating the image, items, and textinput with the channel at hand: [5.3.4 <image>, 5.3.5 <items>, 5.3.6 <textinput>]

In order to keep the RDF and plain XML views of RSS 1.0 in synch as much as possible, RSS 1.0 only supports usage of typed-element RDF syntax ***in the core elements***.

Mime Type
The current mime-type recommendation for an RSS 1.0 document is application/xml. However, work is currently being done to register a mime-type for RDF (and possibly RSS). The RDF (or preferably RSS) mime-type should be used once it has been registered.

File Extension
A specific file-extension for an RSS 1.0 document is not required. Either `.rdf` or `.xml` is recommended, the former being preferred.

Encoding
While RSS 0.9 supported only ASCII encoding, RSS 1.0 assumes UTF-8. Using US-ASCII (i.e. encoding all characters over 127 as &#nnn;) is conformant with UTF-8 (and ISO-8859-1, HTTP's default header encoding).

URLs
As a measure to assure backward compatibility with RSS 0.9, only the following schemes are acceptable in url and link elements: http:, https:, ftp:. mailto: is acceptable in the textinput's link element only.

Entities:
XML reserves certain characters for markup. In order to include these in an RSS document, they must be replaced by their entity reference:

- < becomes `&lt;`
- > becomes `&gt;`
- & becomes `&amp;`

The following two entity references are also recognized by conforming XML parsers. While common, their use is optional. They are, however, required when including a quote character in a string quoted using the same character; e.g. ""Hello," she said" should be encoded as ""Hello," she said".

- ' becomes `&apos;`
- " becomes `&quot;`

Note: Since RSS 1.0 does not require a DTD, be sure to include inline declarations of entities used aside from the aforementioned five. The following DTD fragments are very useful as a source of HTML-compatible entities.

- http://www.w3.org/TR/xhtml1/DTD/xhtml-special.ent
- http://www.w3.org/TR/xhtml1/DTD/xhtml-symbol.ent
- http://www.w3.org/TR/xhtml1/DTD/xhtml-lat1.ent

Usage example:

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
<!ENTITY % HTMLlat1 PUBLIC
 "-//W3C//ENTITIES Latin 1 for XHTML//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml-lat1.ent">
%HTMLlat1;
]>

<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns="http://purl.org/rss/1.0/"
>

...
```

Content Length:
While RSS 1.0 leaves acceptable content length for elements such as title, link, and description to the application, RSS 0.9's maximum character lengths are deprecated to a status of suggested good practice for strict adherence to backward compatibility.

Notation:
In the following core element descriptions, the following notation is used:

- `{something}` is simply a placeholder for a URI, value, etc.
- While, in model descriptions a DTD-like syntax is used, this is for presentation purposes only and does ***not*** imply order. Element order is not important.
- In Model descriptions, ? signifies that an element or attribute is optional.
- In Model descriptions, + means "one or more" instances of this element or attribute is allowed.
- In Model descriptions, * means "zero or more" instances of this element or attribute is allowed.

5.1 <?xml version="1.0"?>

As an XML application, an RSS document is not required to begin with an XML declaration. As a best practice suggestion and to further ensure backward compatibility with RSS 0.9 (the specification for 0.9 required it), this specification recommends doing so.

Syntax: <?xml version="1.0"?>
Requirement: Optional (unless specifying encoding)

5.2 <rdf:RDF>

The outermost level in every RSS 1.0 compliant document is the RDF element. The opening RDF tag assocaties the rdf: namespace prefix with the RDF syntax schema and establishes the RSS 1.0 schema as the default namespace for the document.

While any valid namespace prefix may be used, document creators are advised to consider "rdf:" normative. Those wishing to be strictly backward-compatible with RSS 0.9 must use "rdf:".

Syntax: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns="http://purl.org/rss/1.0/">
Requirement: Required exactly as shown, aside from any additional namespace declarations
Model: (channel, image?, item+, textinput?)

5.3 <channel>

The channel element contains metadata describing the channel itself, including a title, brief description, and URL link to the described resource (the channel provider's home page, for instance). The {resource} URL of the channel element's rdf:about attribute must be unique with respect to any other rdf:about attributes in the RSS document and is a URI which identifies the channel. Most commonly, this is either the URL of the homepage being described or a URL where the RSS file can be found.

Syntax: <channel rdf:about="{resource}">
Requirement: Required
Required Attribute(s): rdf:about
Model: (title, link, description, image?, items, textinput?)
Example:

```
<channel rdf:about="http://www.xml.com/xml/news.rss">
  <title>XML.com</title>
  <link>http://xml.com/pub</link>
  <description>
    XML.com features a rich mix of information and services
    for the XML community.
  </description>

  <image rdf:resource="http://xml.com/universal/images/xml_tiny.gif" />

  <items>
    <rdf:Seq>
      <rdf:li resource="http://xml.com/pub/2000/08/09/xslt/xslt.html" />
      <rdf:li resource="http://xml.com/pub/2000/08/09/rdfdb/index.html" />
    </rdf:Seq>
  </items>

  <textinput rdf:resource="http://search.xml.com" />

</channel>
```

5.3.1 <title>

A descriptive title for the channel.

Syntax: <title>{channel_title}</title>
Requirement: Required
Model: (#PCDATA)
(Suggested) Maximum Length: 40 (characters)

5.3.2 <link>

The URL to which an HTML rendering of the channel title will link, commonly the parent site's home or news page.

Syntax: <link>{channel_link}</link>
Requirement: Required

Model: (#PCDATA)
(Suggested) Maximum Length: 500

5.3.3 <description>

A brief description of the channel's content, function, source, etc.

Syntax: <description>{channel_description}</description>
Requirement: Required
Model: (#PCDATA)
(Suggested) Maximum Length: 500

5.3.4 <image>

Establishes an RDF association between the optional image element [5.4] and this particular RSS channel. The rdf:resource's {image_uri} must be the same as the image element's rdf:about {image_uri}.

Syntax: <image rdf:resource="{image_uri}" />
Requirement: Required only if image element present
Model: Empty

5.3.5 <items>

An RDF table of contents, associating the document's items [5.5] with this particular RSS channel. Each item's rdf:resource {item_uri} must be the same as the associated item element's rdf:about {item_uri}.

An RDF Seq (sequence) is used to contain all the items rather than an RDF Bag to denote item order for rendering and reconstruction.

Note that items appearing in the document but not as members of the channel level items sequence are likely to be discarded by RDF parsers.

Syntax: <items><rdf:Seq><rdf:li resource="{item_uri}" /> ... </rdf:Seq></items>
Requirement: Required

5.3.6 <textinput>

Establishes an RDF association between the optional textinput element [5.6] and this particular RSS channel. The {textinput_uri} rdf:resource must be the same as the textinput element's rdf:about {textinput_uri}.

Syntax: <textinput rdf:resource="{textinput_uri}" />
Requirement: Required only if texinput element present
Model: Empty

5.4 <image>

An image to be associated with an HTML rendering of the channel. This image should be of a format supported by the majority of Web browsers. While the later 0.91 specification allowed for a width of 1-144 and height of 1-400, convention (and the 0.9 specification) dictate 88x31.

Syntax: <image rdf:about="{image_uri}">
Requirement: Optional; if present, must also be present in channel element [5.3.4]
Required Attribute(s): rdf:about

Model: (title, url, link)
Example:

```
<image rdf:about="http://xml.com/universal/images/xml_tiny.gif">
  <title>XML.com</title>
  <link>http://www.xml.com</link>
  <url>http://xml.com/universal/images/xml_tiny.gif</url>
</image>
```

5.4.1 <title>

The alternative text ("alt" attribute) associated with the channel's image tag when rendered as HTML.

Syntax: <title>{image_alt_text}</title>
Requirement: Required if the image element is present
Model: (#PCDATA)
(Suggested) Maximum Length: 40

5.4.2 <url>

The URL of the image to used in the "src" attribute of the channel's image tag when rendered as HTML.

Syntax: <url>{image_url}</url>
Requirement: Required if the image element is present
Model: (#PCDATA)
(Suggested) Maximum Length: 500

5.4.3 <link>

The URL to which an HTML rendering of the channel image will link. This, as with the channel's title link, is commonly the parent site's home or news page.

Syntax: <link>{image_link}</link>
Requirement: Required if the image element is present
Model: (#PCDATA)
Member of: image
(Suggested) Maximum Length: 500

5.5 <item>

While commonly a news headline, with RSS 1.0's modular extensibility, this can be just about anything: discussion posting, job listing, software patch -- any object with a URI. There may be a minimum of one item per RSS document. While RSS 1.0 does not enforce an upper limit, for backward compatibility with RSS 0.9 and 0.91, a maximum of fifteen items is recommended.

{item_uri} must be unique with respect to any other rdf:about attributes in the RSS document and is a URI which identifies the item. {item_uri} should be identical to the value of the <link> sub-element of the <item> element, if possible.

Syntax: <item rdf:about="{item_uri}">
Requirement: >= 1
Recommendation (for backward compatibility with 0.9x): 1-15
Required Attribute(s): rdf:about

Model: (title, link, description?)
Example:

```
<item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
  <title>Processing Inclusions with XSLT</title>
  <link>http://xml.com/pub/2000/08/09/xslt/xslt.html</link>
  <description>
   Processing document inclusions with general XML tools can be
   problematic. This article proposes a way of preserving inclusion
   information through SAX-based processing.
  </description>
</item>
```

5.5.1 <title>

The item's title.

Syntax: <title>{item_title}</title>
Requirement: Required
Model: (#PCDATA)
(Suggested) Maximum Length: 100

5.5.2 <link>

The item's URL.

Syntax: <link>{item_link}</link>
Requirement: Required
Model: (#PCDATA)
(Suggested) Maximum Length: 500

5.5.3 <description>

A brief description/abstract of the item.

Syntax: <description>{item_description}</description>
Requirement: Optional
Model: (#PCDATA)
(Suggested) Maximum Length: 500

5.6 <textinput>

The textinput element affords a method for submitting form data to an arbitrary URL -- usually located at the parent website. The form processor at the receiving end only is assumed to handle the HTTP GET method.

The field is typically used as a search box or subscription form -- among others. While this is of some use when RSS documents are rendered as channels (see MNN) and accompanied by human readable title and description, the ambiguity in automatic determination of meaning of this overloaded element renders it otherwise not particularly useful. RSS 1.0 therefore suggests either deprecation or augmentation with some form of resource discovery of this element in future versions while maintaining it for backward compatiblity with RSS 0.9.

{textinput_uri} must be unique with respect to any other rdf:about attributes in the RSS document and is a URI which

identifies the textinput. {textinput_uri} should be identical to the value of the <link> sub-element of the <textinput> element, if possible.

Syntax: <textinput rdf:about="{textinput_uri}">
Requirement: Optional; if present, must also be present in channel element [5.3.6]
Required Attribute(s): rdf:about
Model: (title, description, name, link)
Example:

```
<textinput rdf:about="http://search.xml.com">
  <title>Search XML.com</title>
  <description>Search XML.com's XML collection</description>
  <name>s</name>
  <link>http://search.xml.com</link>
</textinput>
```

5.6.1 <title>

A descriptive title for the textinput field. For example: "Subscribe" or "Search!"

Syntax: <title>{textinput_title}</title>
Description: Textinput title
Requirement: Required if textinput
Model: (#PCDATA)
(Suggested) Maximum Length: 40

5.6.2 <description>

A brief description of the textinput field's purpose. For example: "Subscribe to our newsletter for..." or "Search our site's archive of..."

Syntax: <description>{textinput_description}</description>
Requirement: Required if textinput
Model: (#PCDATA)
(Suggested) Maximum Length: 100

5.6.3 <name>

The text input field's (variable) name.

Syntax: <name>{textinput_varname}</name>
Requirement: Required if textinput
Model: (#PCDATA)
(Suggested) Maximum Length: 500

5.6.4 <link>

The URL to which a textinput submission will be directed (using GET).

Syntax: <link>{textinput_action_url}</link>
Description: Textinput form action URL
Requirement: Required if textinput

Model: (#PCDATA)
(Suggested) Maximum Length: 500

6. Modules

Namespace-based modularization affords RSS 1.0 compartmentalized extensibility.

The only modules that ship "in the box" with RSS 1.0 are Dublin Core and Syndication, Consult the appropriate module documentation for further information.

Refer to RSS 1.0 Modules for module creation guidelines and registered core RSS 1.0 modules.

Some examples of module usage may be found in the Examples section below.

7. Examples

A basic RSS 1.0 (0.9-like) document, making use of only the core RSS 1.0 element set.

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/"
>

  <channel rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>
      XML.com features a rich mix of information and services
      for the XML community.
    </description>

    <image rdf:resource="http://xml.com/universal/images/xml_tiny.gif" />

    <items>
      <rdf:Seq>
        <rdf:li resource="http://xml.com/pub/2000/08/09/xslt/xslt.html" />
        <rdf:li resource="http://xml.com/pub/2000/08/09/rdfdb/index.html" />
      </rdf:Seq>
    </items>

    <textinput rdf:resource="http://search.xml.com" />

  </channel>

  <image rdf:about="http://xml.com/universal/images/xml_tiny.gif">
    <title>XML.com</title>
    <link>http://www.xml.com</link>
    <url>http://xml.com/universal/images/xml_tiny.gif</url>
  </image>

  <item rdf:about="http://xml.com/pub/2000/08/09/xslt/xslt.html">
    <title>Processing Inclusions with XSLT</title>
    <link>http://xml.com/pub/2000/08/09/xslt/xslt.html</link>
```

```
    <description>
     Processing document inclusions with general XML tools can be
     problematic. This article proposes a way of preserving inclusion
     information through SAX-based processing.
    </description>
  </item>

  <item rdf:about="http://xml.com/pub/2000/08/09/rdfdb/index.html">
    <title>Putting RDF to Work</title>
    <link>http://xml.com/pub/2000/08/09/rdfdb/index.html</link>
    <description>
     Tool and API support for the Resource Description Framework
     is slowly coming of age. Edd Dumbill takes a look at RDFDB,
     one of the most exciting new RDF toolkits.
    </description>
  </item>

  <textinput rdf:about="http://search.xml.com">
    <title>Search XML.com</title>
    <description>Search XML.com's XML collection</description>
    <name>s</name>
    <link>http://search.xml.com</link>
  </textinput>

</rdf:RDF>
```

An RSS 1.0 document pulling in elements from various modules (highlighted in different colours). Note: the modules in this example are for illustrative purposes only; refer to RSS 1.0 Modules for consummate module information.

```
<?xml version="1.0" encoding="utf-8"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
  xmlns:co="http://purl.org/rss/1.0/modules/company/"
  xmlns:ti="http://purl.org/rss/1.0/modules/textinput/"
  xmlns="http://purl.org/rss/1.0/"
>

  <channel rdf:about="http://meerkat.oreillynet.com/?_fl=rss1.0">
    <title>Meerkat</title>
    <link>http://meerkat.oreillynet.com</link>
    <description>Meerkat: An Open Wire Service</description>
    <dc:publisher>The O'Reilly Network</dc:publisher>
    <dc:creator>Rael Dornfest (mailto:rael@oreilly.com)</dc:creator>
    <dc:rights>Copyright &#169; 2000 O'Reilly &amp; Associates, Inc.</dc:rights>
    <dc:date>2000-01-01T12:00+00:00</dc:date>
    <sy:updatePeriod>hourly</sy:updatePeriod>
    <sy:updateFrequency>2</sy:updateFrequency>
    <sy:updateBase>2000-01-01T12:00+00:00</sy:updateBase>

    <image rdf:resource="http://meerkat.oreillynet.com/icons/meerkat-powered.jpg"
/>

    <items>
```

```
      <rdf:Seq>
        <rdf:li resource="http://c.moreover.com/click/here.pl?r123" />
      </rdf:Seq>
    </items>

    <textinput rdf:resource="http://meerkat.oreillynet.com" />

  </channel>

  <image rdf:about="http://meerkat.oreillynet.com/icons/meerkat-powered.jpg">
    <title>Meerkat Powered!</title>
    <url>http://meerkat.oreillynet.com/icons/meerkat-powered.jpg</url>
    <link>http://meerkat.oreillynet.com</link>
  </image>

  <item rdf:about="http://c.moreover.com/click/here.pl?r123">
    <title>XML: A Disruptive Technology</title>
    <link>http://c.moreover.com/click/here.pl?r123</link>
    <dc:description>
      XML is placing increasingly heavy loads on the existing technical
      infrastructure of the Internet.
    </dc:description>
    <dc:publisher>The O'Reilly Network</dc:publisher>
    <dc:creator>Simon St.Laurent (mailto:simonstl@simonstl.com)</dc:creator>
    <dc:rights>Copyright &#169; 2000 O'Reilly &amp; Associates, Inc.</dc:rights>
    <dc:subject>XML</dc:subject>
    <co:name>XML.com</co:name>
    <co:market>NASDAQ</co:market>
    <co:symbol>XML</co:symbol>
  </item>

  <textinput rdf:about="http://meerkat.oreillynet.com">
    <title>Search Meerkat</title>
    <description>Search Meerkat's RSS Database...</description>
    <name>s</name>
    <link>http://meerkat.oreillynet.com/</link>
    <ti:function>search</ti:function>
    <ti:inputType>regex</ti:inputType>
  </textinput>

</rdf:RDF>
```

8. Resources

- Background
  - "RSS: Lightweight Web Syndication"
  - XML Deviant: "RSS Modularization"
  - "Will RSS Fork?"
- RSS
  - Netscape's RSS 0.9 Specification
  - Netscape's RSS 0.91 Specification
  - Netscape's RSS 0.91 Specification, Revision 3
  - Netscape's RSS/MNN Future Directions
  - Userland's RSS 0.91 Specification
  - RSS Usage Survey (25 July 2000)

        - xmlTree's Directory of RSS channels
- RDF & Metadata
        - Resource Description Framework (RDF)
        - W3C Metadata Activity Statement
        - RDFViz
- XML Namespaces
        - Namespaces in XML
- Where to go for more...
        - O'Reilly Network RSS DevCenter
        - RSS Info -- News and information on the RSS format
        - "RSS 1.0: The New Syndication Format"
        - xmlhack
        - XMLfr
- Mailing Lists
        - [RSS-DEV] Mailing List
        - [Syndication] Mailing List
        - [Alchemy] Mailing List

9. Acknowledgements

- The members of the [RSS-DEV], [Syndication] and [RSS] mailing lists for all their continued discussion and input
- The members of the My Netscape Network "Brooklyn" team for RSS 0.9 and 0.91 (Eckart Walther, Jeff Treuhaft, Wade Hennesey, Rafael Cedano, Bill Turpin, Dan Libby, and Mike Homer)
- James Carlyle
- Dale Dougherty
- Edd Dumbill
- Peter Wiggin
- Dave Winer