

# Using RSS for Syndication

*Leigh Dodds*

ldodds@ingenta.com

Research Group, Ingenta plc

3-4 Riverside Court, Lower Bristol Road, Bath, BA2 3DZ Bath, United Kingdom

## Introduction

RSS (RDF Site Summary) is an XML vocabulary that facilitates the syndication of metadata between Web sites. The format originated when large portal sites like My Netscape and My Yahoo emerged, with their personalized view of aggregated headlines from multiple Web sites.

Since that time the RSS format has undergone several revisions in response to its being applied to more varied kinds of applications. Beyond simple headline distribution, RSS is now used to syndicate job listings and discussion group postings, as well as daily content from the personal Web journals known as Weblogs, or simply "blogs." The RSS format has therefore grown to become a format that can be used to syndicate descriptions of all kinds of Web resources.

This article, which introduces the basics of getting started with creating and using RSS-based content, outlines the basic elements in the RSS 1.0 format. It also gives some attention to how best to plan for content syndication.

## Planning for Syndication

A key benefit of syndication is that it allows content to reach a much wider audience.

A "What's New" page on a Web site is useful only for those users who regularly visit the site or stumble across it during a Web search. However, a syndicated version of that same information has a real opportunity to reach a great many more users because the content can flow into a great many more places. Syndication is a valid supplement (or even an alternative) to vying for high-ranking placements in search engines or poorly performing online advertising. By allowing the content to flow to users, syndication provides a means to drive traffic back to a Web site. In short, syndication allows content to break free from the confines of an individual Web site.

When planning for syndication, it's useful to consider two perspectives, your own (i.e., the publisher's) and that of the end user.

From a publisher's perspective the content ripe for syndication is essentially the same content that is probably already highlighted in one way or another on an existing Web site. Press releases, special offers, newly published reports or articles, and new product listings are examples of kinds of promotional content that could be usefully broadcast to a wider audience. Syndication may involve simply extending a content management system or current application to publish existing information in the RSS format as well.

The perspective of the end user should also be considered. Aside from the information that as a publisher you'd like to promote, what information would your end users like to have access to? For example, a software vendor might syndicate a list of common technical support issues or queries. An E-commerce site might allow a user to syndicate a wish list of purchases, favorite authors, or personalized recommendations so as to share this information with friends and family.

Giving users the freedom to access information available from and collected by a Web site is a great way to increase their satisfaction with your service. There's also no loss of control: each resource in an RSS feed is linked back to the originating Web site, encouraging users back to the site in ways that are likely to be more relevant and useful than mere click-through advertising.

Also worth noting is that while syndication through RSS is primarily about plugging into a free, open syndication service, there's nothing to stop metadata syndication being used in more controlled environments (e.g., to drive subscription-based services, to enable exchange of metadata between trading partners, or to be a key component of a corporate intranet). Knowledge-management applications are essentially about delivering relevant information to the users that need it; this is the bread and butter of syndication.



## A Note on RSS Versions

Reflecting its growth and repurposing for new applications, several versions of RSS are in common use. The most popular is RSS 0.91, a very simple format that is eminently suited for simple headline syndication. A more recent development has been RSS 1.0, which was produced in response to a desire for a more modular and extensible format. RSS 1.0 sacrifices some of the simplicity of the RSS 0.91 format, adopting XML Namespaces and an RDF framework to achieve the desired modularity.

The modular nature of RSS 1.0 makes it better suited for extended uses of RSS, allowing publishers to add custom metadata fields to their syndicated content. The following sections introduce the basics of the RSS 1.0 syntax and review some of the standard and proposed modules. (For more detailed information on RSS 1.0 see <http://purl.org/rss/1.0/>.)

## RSS 1.0 Syntax

Before introducing the RSS 1.0 syntax, it's worth highlighting the core data model that an RSS document describes.

An RSS document is a container for two main kinds of resources: a channel and multiple items. Each resource has an associated URL, a title, and an optional description. While a channel can have several additional properties (e.g., an associated image or logo), these three items of metadata are the core properties.

As there are no requirements placed on an item beyond the fact that it is a retrievable resource, there are no limitations on the types of resources that can be described in an RSS document. RSS 1.0 modules allow additional metadata properties to be optionally attached to these different resource types.

Because RSS 1.0 conforms to the RDF specification, this resource-based view of the data model for a channel is prevalent in the syntax. The root element of an RSS document is an RDF container that holds the descriptions of both the channel and the individual items. Metadata in the channel associates the resource items with the channel using a "table of contents."

The following simple document fragment demonstrates the important features of the syntax:

```
<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>
  <!-- channel Table of Contents -->
  <items>
    <rdf:Seq>
      <rdf:li rdf:resource=
        "http://xml.com/pub/2000/08/
          09/xslt/xslt.html"/>
    </rdf:Seq>
  </items>
</channel>
  <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>
</rdf:RDF>
```

There are several things to note about this document.

First, each resource in an RSS document must have an `rdf:about` attribute. This attribute indicates the URL of the resource being described. For a channel this is typically the original URL of the RSS document, or the Web page whose content is being syndicated. For an item this value will be the address from which that item can be retrieved; the value should also match that given in the `link` element associated with each item.

The second point to note about the above example is that both the channel and the item (though only a single item is shown, a channel can contain any number of items) are children of the RDF container element. The association between the channel and the items it contains is described by an RDF sequence (`rdf:Seq`) of one or more list items (`rdf:li`). It



should be obvious from the example that the value of the `rdf:resource` attribute on each list item should correspond to the `rdf:about` attribute of a item described elsewhere in the document.

This trivial example demonstrates the important part of the RSS syntax. (For additional information refer to the RSS 1.0 specification available at <http://purl.org/rss/1.0/spec>. Also, I maintain an online RSS 1.0 validator at [http://www.ldodds.com/rss\\_validator/1.0/validator.html](http://www.ldodds.com/rss_validator/1.0/validator.html) that can be freely used for syntax checking RSS feeds.)

### RSS 1.0 Modularization

To allow RSS 1.0 to be cleanly extended to facilitate syndication of new resource types, modularity has been encouraged through the use of XML namespaces. New properties can be associated with a resource as long as these properties are described by namespace-qualified elements, that is, each module having its own namespace. Extension through namespaces allows a stable core vocabulary to be maintained, while allowing feed publishers to annotate their content with richer metadata that may optionally be understood and processed by aggregators or other consumers.

An RSS module is therefore nothing more than a well-known namespace and a number of associated elements. Three standard modules have so far been produced to supplement the core syntax:

- **Syndication** adds elements containing additional channel level metadata describing the frequency of publication of an RSS feed; the module is used as a hint to aggregators and other applications regularly harvesting the RSS feed.
- **Dublin Core** allows the addition of elements from the Dublin Core Element Set so that richer metadata can be associated with a channel (e.g., publisher) and items (e.g., creator, subject).
- **Content** supplements item metadata with descriptions of multiple formats for content and rich (e.g., XHTML) descriptions for each item.

Members of the RSS community have proposed a number of additional modules, including the following:

- **mod\_audio** associates audio-related metadata with items (e.g., song name, artist, album, bit rate) for syndication of audio play lists.
- **mod\_taxonomy** allows resource items to be classified according to various taxonomies (e.g., the Open Directory Project).
- **mod\_threading** allows the description of parent/child relationships between items to support syndication of discussion group threads.

- **mod\_company** facilitates syndication of news and market data by adding elements for company names, stock symbols, etc.

(A full list of the proposed modules is maintained at <http://purl.org/rss/1.0/modules/>.)

### Plugging into the Syndication Framework

If you have produced an RSS feed, publishing it is simply a matter of making it available from your Web site. Placing a link to the feed from your site will allow users to access the RSS feed from desktop applications such as Headline Viewer (<http://www.headlineviewer.com>), FeedReader (<http://www.feedReader.com>), and Amphetadesk (<http://www.disobey.com/amphetadesk/>). These applications all provide the facility to subscribe to multiple RSS feeds and view them from within a single application.

However, the most important step after creating a public RSS feed is to register it with an aggregator. There are a number of popular RSS aggregators, including Meerkat (<http://www.oreillynet.com/meerkat/index.php>), NewsIsFree (<http://www.newsisfree.com>), Moreover (<http://www.moreover.com>), and the community-run site Syndic8 (<http://www.syndic8.com/>). Aggregators add a great deal of value by

- Collecting feeds into a single easily browsable location
- Correcting problems with individual feeds and providing up/downgrading services to convert between versions of RSS
- Categorizing RSS channels and synthesizing aggregated feeds for these categories
- Providing additional services (e.g., a search engine for RSS content)

Extracting content from the syndication system is simply a matter of finding the feeds of interest—usually from these same aggregators. As an XML application, RSS can be processed with any of the standard tools in the XML developer's toolbox.

### Conclusions

RSS is a simple XML format that provides a framework for online metadata syndication. Syndicating content and data from a Web site provides the opportunity to build new kinds of end user services.

Plugging into the public syndication framework provides the means for this information to reach a much wider audience. The availability of toolkits for publishing and reading RSS-based content makes syndication a very simple process.