



An introduction to RSS news feeds Using open formats for content syndication



James Lewin
President, The Lewin Group
November 2000

Contents:

[What are RSS files?](#)[Creating](#)[Sections](#)[Working with](#)[Publishing](#)[Parsing](#)[Resources](#)[About the author](#)

RDF Site Summary (RSS) is catching on as one of the most widely used XML formats on the Web. Find out how to create and use RSS files and learn what they can do for you. See why companies like Netscape, Userland, and Moreover use RSS to distribute and syndicate article summaries and headlines. This article includes sample code that demonstrates elements of an RSS file, plus a Perl example using the module XML::RSS.

RDF Site Summary (RSS) files, based on XML, provide an open method of syndicating and aggregating Web content. Using RSS files, you can create a data feed that supplies headlines, links, and article summaries from your Web site. These files describe a channel of information that can include a logo, a site link, an input box, and multiple "news items." Other sites can incorporate your information into their pages automatically. You can also use RSS feeds from other sites to provide your site with current news headlines. These techniques let you draw more visitors to your site and also provide them with up-to-date information.

The RSS format originated with the sites My Netscape and My UserLand, both of which aggregate content derived from XML news feeds. Because it's one of the simplest XML applications, RSS found favor with many developers who need to perform similar tasks. Users include Moreover, Meerkat, UserLand, and XML Tree. This article looks at the RSS format and examines some open source Perl modules that will allow you to work with RSS files easily.

What exactly are these RSS files?

RSS files are metadata (see the sidebar [What are metadata?](#)). Until you've used them or seen an example, it may not be easy to understand what RSS files are, but they are easy to create. An RSS file commonly contains four main types of elements: channel, image, items, and text input. These elements are easy to identify and code, as the example in Listing 1 demonstrates. An example of an item within an RSS 0.91 file, Listing 1 contains three easily identifiable parts: a title, a link, and a description.

What are metadata?

RSS files are a type of metadata. Metadata are:

- units of information about information.
- commonly used to provide descriptive information about the content, context, and characteristics of data.

HTML keywords and description metatags are examples of metadata, and are used to provide information about Web pages.

Listing 1. A sample item in RSS

```

<item>
  <title>Mozilla Dispenses with Old, Proprietary DOM</title>
  <link>http://www.mozillazine.org/talkback.html?article=604</link>
  <description>The Mozilla team has decided to forgo backwards compatibility
  with Netscape's proprietary DOM.</description>

```

In headline collections published as results of RSS file aggregations, HTML normally renders the specified title as a headline. The title usually also serves as a link, using the URL listed in the link element. Finally, the description is normally displayed as a summary of the article underneath the headline.

Creating RSS files

You can build RSS files to either the proposed RSS 1.0 specification, or to the currently more popular RSS 0.91 spec. For production applications, use RSS 0.91, because the 1.0 proposal is still under consideration. The [Resources](#) section, at bottom, includes links to both the 1.0 and 0.91 specs. which provide a detailed review of all elements. This discussion focuses on the most commonly used elements, and all the examples in this article are in 0.91 format.

The 1.0 proposal differs from the 0.91 format in one main way: It incorporates Resource Description Framework (RDF) elements that allow greater flexibility at the expense of some increased complexity. This proposed specification is more extensible, creating a W3C standard for RSS files that will meet current needs, will be as backwards-compatible as possible, and will be adaptable to future requirements.

Both versions of the specification share the characteristic of being a lightweight format that developers can use for many purposes.

RSS is an XML application. Because of this, all RSS documents begin with the XML 1.0 declaration followed by the RSS document type declaration, as shown in Listing 2.

Listing 2. The XML declaration

```
<?xml version="1.0"?>
<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS 0.91//EN"
"http://my.netscape.com/publish/formats/rss-0.91.dtd">
<rss version="0.91">
```

The first line declares the document to be an XML document. The second line, the DTD declaration, specifies that this XML file is based on the RSS 0.91 document type definition, DTD, at Netscape. Finally, the root element marks the beginning of the RSS file content, all of which goes between the `<rss version="0.91">` tag and the `</rss>` tag.

The four main sections of an RSS file

After the root element come the four main sections of the RSS file. These are the *channel*, *image*, *item*, and *text input* sections. In practical use, the channel and item elements are requirements for any useful RSS file, while the image and text input are optional.

The channel section

The channel element contains metadata that describe the channel itself, telling what the channel is and who created it. The channel is a required element that includes the name of the channel, its description, its language, and a URL. The URL is normally used to point to the channel's source of information.

Listing 3 shows the beginning of the channel element. This part of the channel element defines the channel and begins the channel information.

Listing 3. The channel element

```
<channel>
  <title>MozillaZine</title>
  <link>http://www.mozillazine.org</link>
  <description>Your source for Mozilla news, advocacy, interviews, builds,
    and more! </description>
  <language>en-us</language>
</channel>
```

The channel element contains the remaining channel tags, which describe the channel and allows it to be displayed in HTML. The title can be treated as a headline link with the description following. The Channel Language definition allow aggregators to filter news feeds and gives the rendering software the information necessary to display the language properly.

The `</channel>` tag is used after all the channel elements to close the channel. As RSS conforms to XML specs, the element must be well formed; it requires the closing tag.

You can include nine optional tags in a 0.91 channel definition. Some examples are PICS Rating, Copyright Identifier, Publication Date, and Webmaster. You can use these additional elements for a variety of purposes. For example, sites that aggregate content can use this additional meta information to allow users to filter news feeds on the basis of Platform for Internet Content Selection (PICS) ratings. For additional information on other Channel tags, look in the RSS specifications.

The image section

The image element is an optional element that is usually used to include the logo of the channel provider. The default size for the image is 88 pixels wide by 31 pixels high, but you can make your logo as large as 144 pixels wide by 400 pixels wide. Here is a sample image element:

Listing 4. The image element

```
<image>
  <title>MozillaZine</title>
  <url>http://www.mozillazine.org/image/mynetscape88.gif</url>
  <link>http://www.mozillazine.org</link>
  <width>88</width>
  <height>31</height>
</image>
```

The image's title, URL, link, width, and height tags allow renderers to translate the file into HTML. The title tag is normally used for the image's ALT text. Keep the image to 88 x 31 or smaller if possible, because many renderers translate channels into fixed width tables as narrow as 100 pixels. Larger graphics could cause the tables to break inappropriately, or cause your image to be left out when displayed.

The items

Items, the most important elements in a channel, usually form the dynamic part of an RSS file. While channel, image, and text input elements create the channel's identity and typically stay the same over long periods of time, channel items are rendered as news headlines, and the channel's value depends on their changing fairly frequently. Here is an example of a channel item:

Listing 5. The item element

```
<item>
  <title>Java2 in Navigator 5?</title>
  <link>http://www.mozillazine.org/talkback.html?article=607</link>
  <description>Will Java2 be an integrated part of Navigator 5?
  Read more about it in this discussion...</description>
</item>
```

Fifteen items are allowed in a channel. This is a reasonable limitation, because most people use channels to distribute recent Web content. Titles should be less than 100 characters, while descriptions should be under 500 characters. The item title is normally rendered as a headline that links to the full article whose URL is provided by the item link. The item description is commonly used for either a summary of the article's content or for commentary on the article. News feed channels use the description to highlight the content of news articles, usually on the channel owner's site, and Web log channels use the description to provide commentary on a variety of content, often on third-party sites.

Much of the beauty of the RSS format lies in the item element. As you can see from the above example, items are easy for developers to define and easy for users to read.

The text input

The text input area is an optional element, with only one allowed per channel. Usually rendered as an HTML form, text input lets the user respond to the channel. You might use this feature to enable your users to subscribe to your newsletter or search your site. Here is an example of a text input element:

Listing 6. The text input element

```
<textinput>
  <title>Send</title>
  <description>Comments about MozillaZine?</description>
  <name>responseText</name>
  <link>http://www.mozillazine.org/cgi-bin/sampleonly.cgi</link>
</textinput>
```

The title is normally rendered as the label of the form's submit button, and the description as the text displayed before or above the input field. The text input name is supplied along with the contents of the text field when the submit button is clicked.

These are the four main elements of an RSS file. After adding the image, item, and text input elements, remember to close the channel with the `</channel>` tag and the RSS file with the `</rss>` tag.

The proposed RSS 1.0 specification introduces modules, which will allow RSS to be extended to accommodate additional information without rewriting the specification. For example, you could write a module to add rich media to your channel for broadband clients while standard clients would still see headlines and descriptions. You may want to learn more about modules so that you can take advantage of them once the 1.0 specification is implemented.

Now start working with RSS files

There are several ways to start working with RSS files. Because RSS files are so simple, they can be created easily using any text or XML editor. Also, there are sites with Web forms that let you create a custom RSS file online. Finally, you will also want to try creating RSS files automatically. Open-source tools for Java, PHP, and Perl can help you get started.

Once you have created a simple RSS file, you will want to validate it. You can do this at Netscape's site, listed below in the [Resources](#) section. Post the RSS file on a publicly accessible area of your Web site, go to Netscape's site, submit your URL, and the validator will test your file for compatibility.

Publishing your RSS file

Once you have a valid RSS file available on your Web site, you can syndicate your content. You can do this in a publish and subscribe fashion -- you publish the information, and anyone who wants to can subscribe -- or you can submit your URL to content aggregators such as Moreover or Userland. Aggregators take content from a variety of sites and publish it in feeds. While your site's information could be mixed in with content from a variety of other suppliers, aggregators can help you dramatically extend the reach of your distribution.

You can also use RSS files for private distribution on intranets or extranets. Their simplicity makes RSS files a good way to publish information to a variety of systems.

Parsing RSS files

Once you start working with RSS files, you will want to parse the file back into discrete units of information. You can do this with the help of a variety of open-source tools written in Java, Perl, PHP, and even ASP. The parser reads a stream of XML text, identifies the opening and closing tags, finds the text enclosed in each tag, and creates handles to work with the parsed information. Once parsed, this information can be incorporated into dynamically generated pages.

Listings 8 and 9 show two simple Perl programs that read RSS files. Even if you don't write Perl, the examples may give you some ideas that you can use in your own development environment.

Perl is a great language for manipulating RSS files; there is a substantial amount of open-source code readily available to help get you started. Jonathan Eisenzopf has developed the `XML::RSS` module, which writes and parses RSS files. To take advantage of this parser, you will also need the `XML::Parser` module. These two Perl modules are available for free at CPAN (see [Resources](#)).

Here is an example of how `XML::RSS` can be used:

Listing 8. A Perl example using XML::RSS

```

# Setup includes
use strict;
use XML::RSS;
use LWP::Simple;

# Declare variables for URL to be parsed
my $url2parse;

# Get the command-line argument
my $arg = shift;

# Create new instance of XML::RSS
my $rss = new XML::RSS;

# Get the URL, assign it to url2parse, and then parse the RSS content
$url2parse = get($arg);
die "Could not retrieve $arg" unless $url2parse;
$rss->parse($url2parse);

```

This code sample passes a URL to a Perl script for parsing. Once parsed, the elements of the RSS file can be used in many ways. For example, you could use RSS items to create a list of headlines:

Listing 9. Making headlines with Perl

```

# Print the channel items
foreach my $item (@{$rss->{'items'}}) {
    next unless defined($item->{'title'}) && defined($item->{'link'});
    print "<li><a href=\"\$item->{'link'}\">$item->{'title'}</a><BR>\n";
}

```

This sample loops through the array of RSS items, verifying that each item comes complete with a title and link. Incomplete items are skipped; complete items are included in a list of linked headlines.

If you plan to use the XML::RSS module, open and read it with any text editor; it is heavily commented with suggestions for using it effectively.

Once you have tried your hand at RSS files, you'll find that there are many ways that you can use them. For example, you can write scripts that generate RSS summaries every time your site is updated, or scripts that periodically retrieve news from other sites and automatically update your own news page. (How to write those scripts is fodder for another article, but you may find some useful open-source tools to automatically generate RSS summaries in the tool sources listed in [Resources](#).

I've offered a few suggestions for creating and using RSS files. The resource section provides additional information, such as sources for RSS files, the RSS specifications, and places where you can post your headlines.

Resources

Related standards

- The [RSS 1.0 Specification Proposal](#) site contains general information such as background, motivation, and design goals as well as the working specification.
- Netscape's RSS 0.91 is part of their [Quick Start Guide](#) to RSS and provides a step-by-step approach to creating your own My Netscape channel.
- The W3C Recommendation for the RDF model and syntax specification is contained at [Resource Description Framework](#).

Online RSS file editor

- Webreference has an online [RSS Channel Editor](#) that is a great way to get started making RSS files.

RSS software tools

- CPAN has the latest versions of the Perl parser and RSS modules at [CPAN List](#).
- ASP tools for RSS can be found at [TNL Net](#).
- [Wireless Developer Network](#) has tools for parsing RSS files with PHP.

RSS news sources: Syndicate your information, or get news feeds

- [Moreover](#) is an aggregator that features free news feeds from over 1,500 news sources.
- Netscape originated the format as RSS 0.9. Their site features an overview of RSS 0.9 and has a place for [listing your site's channel](#).
- For examples of syndicated content, take a look at [XML-Tree](#), which has a large collection of channels.
- [My Userland](#) aggregates headlines from a variety of sources. It was one of the first sites to use RSS files.
- [Meerkat](#) is an RSS-based syndicated content reader, as well as a source for news feeds.

About the author

James Lewin has been working with the Internet since 1995. He is the president and owner of [The Lewin Group](#), a networking and Internet solutions provider. He is an MCSE who also works with Microsoft and open-source Internet development tools. He can be reached at jim@lewingroup.com.

What do you think of this article?

Killer! (5)

Good stuff (4)

So-so; not bad (3)

Needs work (2)

Lame! (1)

Comments?

[Privacy](#) [Legal](#) [Contact](#)