RSS for Beginnners

Have you ever noticed those inviting orange buttons on some web pages, or spotted the odd link pitching an "RSS feed"? If you've ever clicked one out of curiosity, and then scratched your head at the unformatted gobbledygook in your web browser, you've seen an RSS file.

What is it really for, anyway?

Two things: RSS (Really Simple Syndication) and Atom are two specialized formats that create what's commonly called a news feed, or just feed. A feed is an easy way for sites to share headlines and stories so that you can read them without having to visit two dozen different web pages everyday.

In other words, web builders use feeds to dish out fresh news and content from their websites and web surfers can use feed applications to collect custom-tailored selections of their favorite websites to be read at their leisure.

What we'll cover

Because there are several different formats — RSS and Atom are the main ones — we'll simple refer to them as feeds.

The idea of feeds — pushing out your content to users, rather than forcing them to visit your site — gained traction among developers years ago, in part because they're really easy to code and even simpler to share.

This brief how-to will get you started on the road to feed consuming nirvana and we'll also take a quick look at creating your own no-frills RSS feed – one you can build for your site in just a few minutes.

Consuming feeds

Feed reading apps bring a wide selection of bespoke news to your desktop without requiring you to wade through links and bookmarks. Generally free of layout code, heavy graphics, and advertising, RSS feeds download quickly. Plus, RSS uses simple code, so it's available no matter what kind of device you're using. For example, I often lack access to telephone lines when I'm on the road, so I use my Bluetooth cellular modem and NetNewsWire to keep in touch with my favorite sites.

Most modern web browsers offer a built in feed reader with varying features. Although browser-based feed readers may not be the most sophisticated way to read your favorite sites' feeds, they make a simple place to start.

When you click on one of those lovely little orange icons in Firefox, Safari, Opera or IE7 you'll likely find yourself starring at a somewhat raw looking page that shows all the headlines, publication dates and short summaries of stories that the site has recently published.

That, in a nutshell, is your feed. If you add this feed to your browser's feed section (often referred to as "live bookmarks" you can return tomorrow and find any updates from the site waiting for you.

That's all well and good, but there's a lot more you can do with feeds. Once you start to see the brilliance of reading the news through feeds, you'll likely want a more sophisticated means of organizing and reading your feeds.

There are a number of desktop clients available of all the major platforms and most of them are free. Popular apps include NetNewsWire (Mac), FeedDemon (Windows) and Straw (Linux). If you don't want a dedicated feed reader, many e-mail clients also offer support for new feeds.

Then there's the ever-popular web-based solutions like Google Reader, Bloglines, Netvibes and countless others. The web-based options often have nice sharing features, like Google Reader which allows you to share items from your news feed with others in a new news feed — feeds within feeds within feeds.

Publishing feeds

From a publishers standpoint feeds sound like a nightmare since they let consumers see your content without visiting your site. That means they get all the good stuff, but you don't get any page views and advertising hits.

Such thinking fails to see the whole picture though. Compared to the richness of a website, feeds offer little more than text. Feeds offer your users a way to focus on what matters to them, ignoring what doesn't. At the same time, when your readers do find something they like, we can almost guarantee they'll click through to your site. Many of them will want to bookmark the page, which requires a visit to your site and if your content is compelling enough they may even link to what you wrote from their own sites.

Offering RSS or Atom feeds will increase your traffic because it gets your content out in the larger world where more people can find it. Feeds are good for traffic and don't let anyone convince you otherwise.

You don't need to be the AP Newswire or an online Samuel Pepys to get good use out of RSS. RSS can spread the word about a band's tour dates, corporate league sports schedules, civic functions, snow reports, real estate listings, university lectures, software updates, et cetera. If you

semi-regularly update content on your website, building an RSS feed gives you another worthwhile way of delivering your content to your readers.

Feeds, particularly Atom feeds, though RSS will work as well, can also serve as kind of primitive API that allows your industrious readers to grab your data and do interesting things with it. Perhaps you have a byline that includes where you published from, users can pull out that data, run it through something like Yahoo Pipes and create a mashup that displays your posts on a map.

Obviously a feed is not a substitute for a full-fledged API, but you may be surprised what some of your more creative readers/programmers are able to do with your feeds.

Producing your own RSS and Atom feeds

Most of today's more popular publishing systems — WordPress, Movable Type, Blogger, LiveJournal, etc. — have built in ways to generate RSS or Atom feeds. All you need to do is add a link on your pages.

More sophisticated options, like a feed for an entry's comments (so people can follow your discussion from afar) will require a plugin or some custom hacking.

If you're using your custom built site you may need to write your own RSS or Atom feeds.

But don't worry, RSS is pretty simple to handle and, while Atom is a little more complex, it isn't too hard either.

The history of RSS is convulted and somewhat boring, here's the ten second summary: There are 3 revisions/version of RSS with slightly different features in each. RSS has, at various points stood for Rich Site Summary (RSS 0.91), RDF Site Summary (RSS 1.0 and RSS 0.90) and Really Simple Syndication (RSS 2.0).

Of these RSS .91 is probably the most widely supported format, though RSS 2.0 is quickly becoming the standard. For simplicity and forward-compatibility we'll stick with RSS 2.0.

Inside a typical feed

Here's a sample version of an RSS 2.0 feed taken from my Flickr feed (I cut out a few things, but otherwise it's the same for any Flickr user):

```
01 <?xml version="1.0" encoding="utf-8"?>
02
03 <rss version="2.0"
04
05      xmlns:media="http://search.yahoo.com/mrss/"
06
```

```
07        xmlns:dc="http://purl.org/dc/elements/1.1/"

08

09        xmlns:flickr="http://flickr.com/services/feeds/">

10

11    <channel>

12

13        <title>Uploads from luxagraf</title>

14

15        <link>http://www.flickr.com/photos/luxagraf/</link>

16

17        <description></description>

18

19        <pubDate>Thu, 1 May 2008 17:03:32 -0800</pubDate>

20

21        <lastBuildDate>Thu, 1 May 2008 17:03:32 -0800</lastBuildDate>

22

23        <generator>http://www.flickr.com/</generator>

24

25        <image>

26

2           <url>http://farm1.static.flickr.com/25/buddyicons/85322932@N00.j
7 pg?1181220289#85322932@N00</url>

2
8

29          <title>Uploads from luxagraf</title>

30

31          <link>http://www.flickr.com/photos/luxagraf/</link>

32

33        </image>

34

35        <item>

36

37          <title>texture 1</title>

38

39          <link>http://www.flickr.com/photos/luxagraf/2457443509/</link>

40

            <description><p><a
41 href="http://www.flickr.com/people/luxagraf/">luxagraf</a> posted a
   photo:</p>

42

43          <p><a href="http://www.flickr.com/photos/luxagraf/2457443509/"
```

```
        title="texture 1"><img
        src="http://farm3.static.flickr.com/2287/2457443509_7b85441d35_m.jpg"
        width="180" height="240" alt="texture 1" /></a></p>
44
45              </description>
46
47              <pubDate>Thu, 1 May 2008 17:03:32 -0800</pubDate>
48
49          </item>
50
51      </channel>
52
53 </rss>
```

What do you see? Some funny-looking tags, perhaps, but it's a lot like HTML and XML, isn't it? Since RSS is an application of XML, RSS feeds must be built as well-formed XML — so when you open a tag, remember to close and nest it properly. Sloppy code won't work here.

The XML rigmarole likewise requires the first two lines, first defining the XML version. If you don't know what that means, no worries. Just cut and paste those opening lines, they're the same for every 2.0 feed.

Next are a few lines specify the RSS version and providing links

Then you'll notice the `channel` tag. An RSS channel is the container for each of the items you're going to publish. The channel metadata then consists of general information about your site, or metadata. Roughly akin to the `<head>` of an HTML file, this portion of the feed stays the same as updates are added. Here's where you can start replacing tidbits to tailor it to your site.

Here's tag-by-tag commentary:

`<RSS>`: This opening tag includes a mandatory version attribute. Note that the `</rss>` tag also concludes our feed.

`<channel>`: The channel is the fundamental container for all RSS data – there's only one channel in a feed. Note that the channel tag gets closed near the very end of the feed, too.

`<title>`: Hey, the title! This is most likely going to be the same title as your homepage.

`<link>`: The URL for the webpage that corresponds to the RSS feed. (Most likely, this is your homepage's URL.)

`<description>`: A brief description of what's in this feed, or the purpose of your site.

`<pubDate>` and `<lastBuildDate>`: PubDate refers to the publication date for the content in the channel. For example, if you publish on a daily basis, just update this date once every 24 hours. LastBuildDate refers to the last time the content of the channel changed. In other words whenever you add new content to your feed, update this timestamp. And keep in mind that all date-times in RSS conform to the Date and Time Specification of RFC 822.

`<generator>` and `<image>`: Generator just refers to who or what created the file and image is a (totally optional) tag to specify an image that goes with the feed (in Flickr's case it's your user image).

Now, onto the `<item>`s – the dynamic headlines, links, and content you'll be syndicating. When you update your site and add new stories, new items are added to the RSS feed. Each `<item>` represents a separate story or content update. Up to 15 items can be included in RSS 0.91.

`<item>`: This wrapper tag is required around every item

`<link>`: The permantent URL of an item.

`<description>`: A synopsis or excerpt of the item, although you're free to publish the entirety of the item here, as is many peoples' practice.

`<pubDate>`: The date for the individual item.

Two pieces of advice to heed while creating your `<description>`: Firstly, put in the extra effort to create a well-written, easy-to-read description. In the text-centric world of RSS, you can't expect an audience to click through to your site if your "teaser" excerpt has little appeal or sense to it. Secondly, if there is HTML code in your description, XML parsers throw a fit unless certain HTML symbols (like the ampersand) are escaped out. Either keep HTML out of your description altogether, or encode it via CDATA, like so:

```
1 I would <![CDATA[<b>really</b>]]> rather have just dropped the bold tag.
```

And that's it. Done!

Next, let's validate and automate.

Validating feeds

To ensure your feed is properly formatted, run it against an RSS validator. The validator provided by Feed Validator is a good place to start.

Our example on the previous page is a bare-bones RSS feed. There are plenty of other, optional tags you can include if you wish. You can specifying things like language, copyright and more. Be sure to real the official specification for RSS 2.0 to get the full details.

Of course, the RSS file you've created needs to be updated each and every time you add content your site. To avoid the dull and error-prone task of updating RSS feeds manually, most web builders use automated tools to update their RSS.

There are resources on the web which hand out free scripts and tools that you can use to spice up your RSS. webreference offers a web-based form that helps automate the RSS-building process. webdevtips.com has a nice web form, too. Additionally, a number of simpler server-side scripts to create RSS are available in PHP or another language of your choice.

Now that your RSS feed is all dressed-up, let's announce its availability to the world. Debutante ball, anyone?

Publicize your feed

After uploading your RSS file to your server, you'll still need to inform people of its existence. When you create a new feed, running through these few steps will debut your feed in style:

- Advertise to web surfers! Put an XML button or text link on your page, linking to the RSS file. This [XML] button is the widely-recognized graphic for RSS feeds, though variants and remixed designs are generally just as recognizable. Go ahead, right-click, Save as…
- There is also a site fully dedicated to standard Newsfeed icons (highly recommended):
- Advertise to the machines! Most RSS applications (and some search spiders) will automatically determine your feeds' location when you put the following code in the `<head>` section of your homepage: <link rel="alternate" type="application/rss+xml" title="RSS" href="url/to/rss/file" />
- Get listed by the major feed directories! Syndic8.com and News Is Free are two of the biggest collections of RSS feeds. Before advertising yourself to these sites, though, run a final test against an RSS validator. While web browsers will render many poorly-coded web pages, RSS parsers can be less forgiving, and require a well-formed XML file to work with.

Of course, there's always more to discover, if you are so inclined. Solid RSS tutorials have been written by Mark Nottingham, Fagan Finder, and Danny Sullivan. We'd be remiss not mentioning webReference.com's large RSS library, and the surprise entry from the State of Utah which provides a great introduction to the subject. Lastly, of course, there's a definitive dead-tree resource from O'Reilly press, covering RSS in depth, not to mention a special RSS DevCenter site.

Happy feeding!