

# Implementing RSS using PHP

by Craig Murphy

**RSS is a TLA that has crept into usage without very much fuss at all. Precisely what RSS stands for is still a matter of debate, albeit not a big debate. Some people think it stands for “Really Simple Syndication”, others think it as “RDF Site Summary” or “Rich Site Summary”. Me? Well I believe it stands for Really Simple Syndication – not because it is simple, but because it is concise. Over the course of this article I will briefly explain RSS and will demonstrate how I incorporated the UK BUG’s RSS feeds into my own web-site. Whilst I will be using PHP in this article, there are C# and Delphi resources that can achieve similar results.**

## Preamble

If you want to learn about a new technology or explore a new language, nothing focuses the mind more than writing an article or giving a presentation – I can thoroughly recommend it. That’s what has happened here – our editor wondered if I would add information about the BUG and DDG MasterClasses to my web-site.

When Joanna first asked me to include the MasterClass information on my site, my first reaction was “Ah! Web service!” Even Dr. Bob was keen to present me with the information in whatever format I required. After a sending a few e-mails back and forth, it suddenly became clear that one of Dr. Bob’s earlier prophetic visions included storing information about the MasterClasses in an XML format known as RSS. Thinking about the web service idea a little more, we would have probably ended up with an XML format whether the RSS format existed or not.

Luckily, I knew a little about XML and how to work with it, but didn’t know much about RSS – I had heard of it and knew roughly what it could do. Dr. Bob updates the two RSS files (one for BUG, the other for DDG), so it made sense to make use of those files. After all, think of the chain of updates that would have to occur if I simply cut’n’paste’ the MasterClass information whenever Joanna updated it. Whenever a new MasterClass is available, Joanna has to advise Dr. Bob to update the RSS files; Joanna then has to let me know that the MasterClass information has changed...it is a little convoluted, I am sure you get the picture!

Would it not be nice if my site updated itself whenever Dr. Bob changed his RSS files? Well, that’s the spirit of RSS: hands-free updates. To that end, I now have a very small PHP script that “pulls” the RSS files from <http://www.dotnetuk.com/DDG.xml> and <http://www.richplum.com/BUG.xml> and presents them on my site. You can visit the URL here: <http://www.craigmurphy.com/bug/training.php>.

The remainder of this article will outline how I put training.php together. This is not intended to be a PHP tutorial, however I will explain one or two of the issues that PHP raises when using it out of the box.

## What is RSS?

RSS is an XML format that can be used to describe almost anything. RSS files or “feeds” are simply text files, often with a .rss or .xml extension. Typically RSS is used for publishing news items on the Internet. However its usage is not limited to that. For example, it is possible that you might use RSS to advise your application that an update is available – your application could poll a particular RSS file (on your web-site) looking for a message that a new version exists. Similarly, you could use RSS to publish your Frequently Asked Questions (FAQs) – your application would always have the most up-to-date FAQs.

Another such use for RSS might be the publication of an e-book. Imagine you are writing and releasing a chapter per week – an RSS feed is an ideal method for announcing when each chapter is available. RSS can effectively do away with the “check this site frequently for updates” that so many web sites announce.

The BBC news web-site offers an RSS feed – this is simple an XML file that contains the same headlines that you would see were you looking at the BBC web-site or viewing the BBC news ticker. This raises the subject of push and pull. By viewing the BBC web-site *you* are pulling information to you. In contrast, the BBC news ticker pushes information to you – and it does this as soon as the BBC update their news items.

RSS offers a means of allowing you to pull information to your PC as and when you like – and its key benefit is the fact that it can be automated. This is in contrast to the BBC news web-site which you must visit using your browser, i.e. you must choose when to refresh the news information, by which time the information could be hours old.

Sites that offer RSS feeds usually have an XML or RSS icon hidden somewhere. Figure 1 highlights the RSS icon on the Richplum web-site.

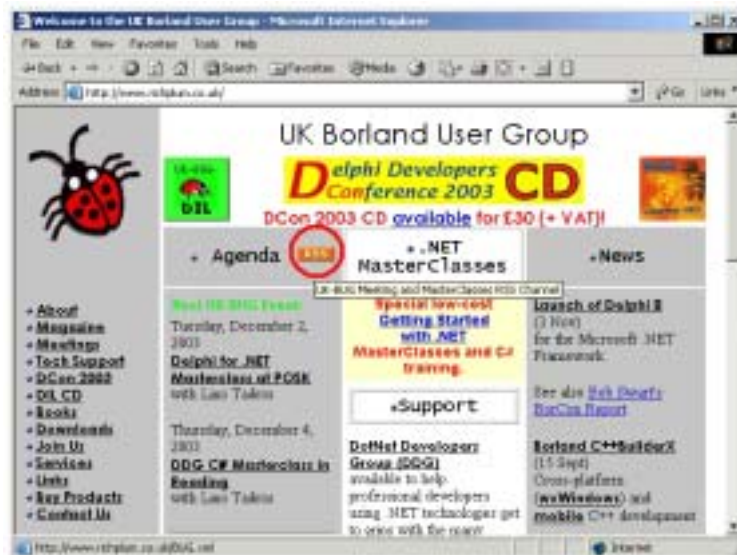


Figure 1 – The RSS Icon

I will say this: be very careful with RSS feeds – they have the potential to consume huge amounts of time!

## Viewing RSS feeds

Before we get into the details of what RSS looks like and how we can use it, let us consider RSS from a user's perspective. Applications that "read" RSS files are known as RSS aggregators – they bring together the information in one or more RSS files and present it to the user in a readable and manageable format.

I use a .NET program call RSSReader (there is a link in the RSS Aggregators section at the end of this article) to view the RSS feeds from sites that interest me. Figure 2 presents a screenshot of RSSReader at work – in this screenshot I am viewing the DDG MasterClasses RSS feed and have selected Lino's C# MasterClass held in Reading. This particular RSS viewer will allow me to "Read More" in the same window that the brief RSS item is display in, or it will let me open the article in my browser.

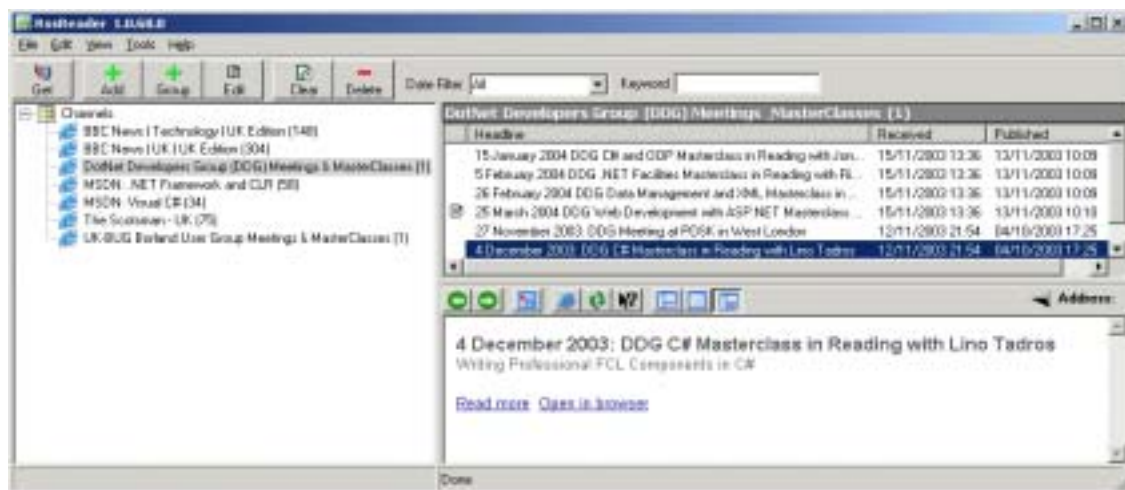


Figure 2 – Viewing an RSS file/feed

## No More Screen-Scraping?

One of the primary advantages of RSS is the ease in which it can be incorporated into your own applications and web-sites. With the advent of RSS, "screen-scraping" is no longer a necessary evil. Assuming widespread adoption of RSS, those sites with useful information will publish that information via an RSS feed.

Screen-scraping is the process of automated information extraction, typically from an HTML page, making assumptions about what information can be found on the page and where it is located on the page. It is a troublesome process and one that frequently fails when the page being scraped is re-designed. Amazingly, I have seen a number of web services that use screen-scraping as their means of gathering information!

## An Example...BUG/DDG MasterClasses

Listing 1 presents the RSS file (or feed) for the BUG MasterClasses. As you can see, RSS feeds are a readable XML format. Inside the root node, <rss>, there is a <channel> node that contains a collection of <item> nodes and a number of auxiliary one-off nodes, such as <managingEditor>.

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>UK-BUG Borland User Group Meetings & MasterClasses</title>
    <link>http://www.ukbug.co.uk</link>
    <description>UK-BUG is an association of professional developers using Borland Development Tools. The aim of the group is to provide members with essential information and training as economically as possible in terms of both time and money.</description>
    <language>en-uk</language>
    <copyright>Copyright 2003 by Richplum Ltd.</copyright>
    <managingEditor>b.swart@chello.nl (Bob Swart)</managingEditor>
    <webMaster>b.swart@chello.nl (Bob Swart)</webMaster>
    <pubDate>2003-11-07 13:27:43</pubDate>
    <lastBuildDate>2003-11-07 13:27:43</lastBuildDate>
    <generator>eBob42RSS v0.42</generator>
    <docs>http://blogs.law.harvard.edu/tech/rss</docs>
    <image>http://www.ukbug.co.uk/images/UKBUGS.gif</image>
    <item>
      <title>4 December 2003: DDG C# Masterclass in Reading with Lino Tadros</title>
      <link>http://www.richplum.co.uk/meetings/20031204.asp</link>
      <description>Writing Professional FCL Components in C#</description>
      <author>Bob Swart (aka Dr.Bob - b.swart@chello.nl)</author>
      <comments>http://www.ukbug.co.uk/BUG.xml</comments>
      <pubDate>2003-11-07 13:27:41</pubDate>
    </item>
    <item>
      <title>2 December 2003: Delphi for .NET Masterclass at POSK with Lino Tadros</title>
      <link>http://www.richplum.co.uk/meetings/20031202.asp</link>
      <description>Fun Day with Delphi for .NET</description>
      <author>Bob Swart (aka Dr.Bob - b.swart@chello.nl)</author>
      <comments>http://www.ukbug.co.uk/BUG.xml</comments>
      <pubDate>2003-10-09 22:50:27</pubDate>
    </item>
  </channel>
</rss>
```

**Listing 1 – BUG MasterClasses pulled from <http://www.richplum.com/BUG.xml>**

## Using RSS with PHP

Rather than re-invent the wheel, there just had to be an RSS component that I could download, install and make use of. True enough, there was: MagpieRSS provides an XML-based (expat) RSS parser in PHP (there is a link in the Resources section of this article). I downloaded MagpieRSS and did nothing more than copy it in to the 'rss' directory in the root directory of my web server (in my case this is IIS, c:\inetpub\wwwroot\rss).

Given that I am still using a dial-up connection, I downloaded Dr.Bob's BUG.xml and DDG.xml RSS files and saved them in the 'rss' directory. This was enough to let me test MagpieRSS locally. I then wrote listing 2. I say wrote, I actually mean cut'n'paste from the MagpieRSS examples.

```
<?php
require_once 'rss/rss_fetch.inc';

// Go live: Replace with http://www.richplum.com/BUG.xml
$url = 'http://localhost/rss/BUG.xml';
$rss = fetch_rss($url);

echo "<h1>Site: ", $rss->channel[title], "</h1><br><br>";

foreach ($rss->items as $item ) {
  $title = $item[title];
  $url = $item[link];
  $desc = $item[description];
  echo "<a href=$url>$title</a></li><br>\n";
  echo "$desc<br>\n";
}

?>
```

**Listing 2 – Using BUG.xml locally**

Listing 2 uses MagpieRSS's fetch\_rss function to retrieve BUG.xml. It then outputs the channel/title node as vanilla HTML. The foreach statement iterates over the rss/channel/item nodes (of which there are two in the BUG.xml RSS feed) outputting the title (as an HTML anchor), link and description as HTML.

PHP lets us treat the rss/channel/item nodes as if they were arrays. PHP also lets us access the information within each item element by meaning names that map onto the XML node names – title, link, description, author, comments and pubDate. This flexibility, whilst good for code readability, is not without a downside as we shall see now.

## PHP Warnings/Notices

If you try out listing 2, you may notice [sic] that the resultant web page contains PHP processor warnings or notices. This is perfectly normal – the associative array keys, title, link, and description are not really defined anywhere, the PHP interpreter is warning us of that fact. PHP stores a lot of internal configuration in a file called php.ini – it is possible to disable these notices or warnings by adjusting the error\_reporting variable in php.ini. On the Windows platform, php.ini lives in the C:\Windows directory. However, this will affect all PHP applications/scripts, which might not be a good thing. Therefore I suggest adjusting the error\_reporting variable on an as and when required basis. This can be achieved using the following line of PHP:

```
error_reporting (E_ALL ^ E_NOTICE);
```

## Putting It All Together

If we ran Listing 2 against both BUG.xml and DDG.xml, Figure 3 should look familiar.



Figure 3 – Using Listing 2

It is worth noting that I have not used Listing 2 in the main “index.php” page at <http://www.craigmurphy.com>. This is purely a safety net – if either of the RSS feeds from <http://www.richplum.com> or <http://www.dotnetuk.com> were unavailable, any pages requesting the RSS feeds would appear to stall whilst a timeout occurred. For this reason I tend to move pages that reference other sites *off* my main page.

However, once we have the RSS items, it is possible to use any look’n’feel we choose. Figure 4 presents a screenshot of the output from <http://www.craigmurphy.com/bug/training.php>.



Figure 4 – The final product

## Summary

Hopefully this whistle-stop look at RSS has given you an insight into what RSS is doing for the Group. Maybe, after reading this article, you have thought of some subtle new uses for RSS – if you have please share them with us either in this publication or in the Richplum newsgroups.

Lastly, if you are interested in learning more about RSS, O'Reilly publishes "Content Syndication with RSS" by Ben Hammersley – thankfully it is a slim volume!

## Resources

- Magpie RSS: [http://sourceforge.net/project/showfiles.php?group\\_id=55691](http://sourceforge.net/project/showfiles.php?group_id=55691)
- RSS at XML.COM: <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>
- Google's RSS resource:  
[http://directory.google.com/Top/Reference/Libraries/Library\\_and\\_Information\\_Science/Technical\\_Services/Cataloguing/Metadata/RDF/Applications/RSS/News\\_Readers/](http://directory.google.com/Top/Reference/Libraries/Library_and_Information_Science/Technical_Services/Cataloguing/Metadata/RDF/Applications/RSS/News_Readers/)
- An index of over 10,000 RSS feeds can be found here: <http://www.syndic8.com/>

## RSS Aggregators:

- RSSReader (requires .NET): <http://www.rssreader.com>
- FeedReader: <http://www.feedreader.com/>

---

*Craig is an author, developer, speaker and Dilbert Evangelist – he specialises in all things XML, particularly SOAP and XSLT. Craig is evangelical about .NET, C#, Test-Driven Development, Extreme Programming and Scrum. He can be reached via e-mail at: [Craig@isleofjura.demon.co.uk](mailto:Craig@isleofjura.demon.co.uk), or via his web site: <http://www.craigmurphy.com> (where you can also find the source code and PowerPoint files for all of Craig's UK-BUG articles, reviews and presentations).*