

## PHP/MySQL RSS Feed Tutorial

---

If you have a web site that is database driven with regularly updated material then it may be a good idea to setup an RSS feed for your website. This tutorial assumes you have a basic understanding of PHP and MySQL — otherwise how else did you get a database driven web site? ;)

The first step is to use PHP to declare an XML document.

```
<?php
header("Content-type: text/xml");
echo "<?xml version='1.0' encoding='UTF-8'?>";
```

The `'header'` sets the HTTP header to text/xml so the client (RSS reader) knows that the content-type is XML. The rest of the PHP sets the document type as XML version 1.0. The next step is to set the RSS version.

```
// Set RSS version.
echo "
<rss version='2.0'>";
```

Just in case you're wondering why there is a line break after the `'echo "'`, it is to make the XML document code neater when it's generated. The same system is used throughout this tutorial.

Next, we can begin to create the actual XML document.

```
// Start the XML.
echo "
<channel>
  <title>Feed title</title>
  <description>A description of the feed contents</description>
  <link>http://www.yoursite.com/</link>";
```

What the above code does is create the opening `'<channel>'` tag as well as the required `'<title>'`, `'<description>'` and `'<link>'` elements.



## PHP/MySQL RSS Feed Tutorial

---

Once the start of the document has been setup, it's time to get the data that will fill the XML document from the database. I have a PHP file created that I use for all my database connections that I link to using the following.

```
// Create a connection to your database.  
require("db_connection.php");
```

The code that you would find on that page would be like this:

```
<?  
@mysql_connect('server', 'username', 'password') or die('ERROR--CAN'T CONNECT TO  
SERVER');  
@mysql_select_db('database_name') or die('ERROR--CAN'T CONNECT TO DB');  
?>
```

The first line (`@mysql_connect...`) connects to the server while the second line (`@mysql_select_db...`) connects to your database. You need to replace the relevant information with your own server/database details for this to work.

Now we can begin to add the data that gets pulled from the database.

```
// Query database and select the last 10 entries.  
$data = mysql_query("SELECT * FROM table ORDER BY id DESC LIMIT 10");  
while($row = mysql_fetch_array($data))  
{  
    // Convert database images data into actual image link.  
    $row[Intro] = str_replace("images/", "http://www.yoursite.com/images/",  
    $row[Intro]);
```

What we have done now is query the database and selected everything from the table named `'table'` but limited the results to the last 10 (ordered latest to oldest by their id). You would need to edit `'table'` to the name of the table you want to create a feed from.

`'$row[Intro]'` assumes you have a column in your table called `'Intro'`. You would need to rename this to whatever the column name is that contains the data you want to



## PHP/MySQL RSS Feed Tutorial

---

have displayed in the RSS reader.

The `str_replace` function replaces all instances of `'images/'` with `'http://www.yoursite.com/images/'` which is very important if the column you are getting data from in your database contains links to images on your site. If this is not the case, feel free to ignore that line of code. And if you have a different location for your images, edit accordingly.

The next part is where the XML document is filled with data.

```
// Continue with the 10 items to be included in the <item> section of the XML.
```

```
    echo "  
    <item>  
        <link>http://www.yoursite.com/article.php?id=".$row[id]."</link>  
        <guid isPermaLink=\"true\">http://www.yoursite.com/article.  
php?id=".$row[id]."</guid>  
        <title>".$row[Title]."</title>  
        <description><![CDATA[\". $row[Intro].\"]></description>  
        <comments>http://www.yoursite.com/article.php?id=".$row[id]."#Comments</  
comments>  
    </item>";  
}
```

The code above creates the `<item>` and its required `<link>`, `<title>` and `<description>` elements. I have also included the `<guid>` element as it is recommended by the [Feed Validator](#). The `<guid>` element uniquely identifies the item. In my case, the link URL is always unique and therefore works well as the `<guid>`. If the link to the article will always remain the same, then set `'isPermaLink'` to `'true'`. If the URL will change, set it to `'false'`. You will need to edit the `<link>` and `<guid>` elements to display whatever web pages your site uses.

The contents of the `<link>` element contain the URL of the full document you are linking to. `<title>` displays the title of the document (if your title field is not named `'Title'` then rename `'$row[Title]'` to suit your database).

The `<description>` element contains the text from the article that you want to have



## PHP/MySQL RSS Feed Tutorial

---

displayed in the RSS reader. An important part of this element's contents is the **CDATA** section. Normally, XML parses all of the text contained within it. So if your text contains characters like '<' or '&' then your description will be broken. If you plan on including images in your feed, then you'll need to include the **CDATA** code.

If the text you are including in your description can be very lengthy you may also want to consider using `substr`. So instead of displaying `<description><![CDATA[".$row[Intro]."]]></description>` you would show `<description><![CDATA[".$row[Intro],0,150)."]]></description>`. What the `substr` code does is start from the first character (represented by the '0') and counts to the 150<sup>th</sup> character (represented by the '150'). Anything after the 150<sup>th</sup> character is not displayed. Again, feel free to change the '150' to any number that suits your needs.

Finally, another optional element I have included is the `<comments>` element. If the URL of your article allows for comments you can include a link to where the comments can be found.

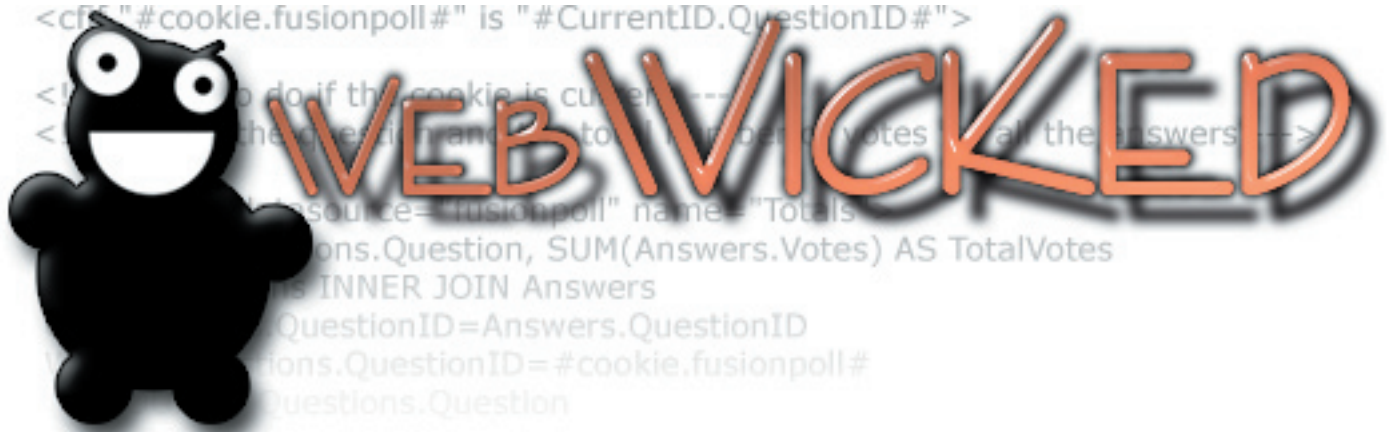
To get a full description of all the different tags available in RSS, check <http://feedvalidator.org/docs/>.

All that is left to do now is close the `<channel>` element and RSS document using PHP.

```
echo "  
    </channel>  
</rss>";
```

```
?>
```

You have now created an XML document/RSS feed. I have included the completed code on the following page. Please note, this tutorial does contain formatted text which you will need to replace if you plan on copying and pasting it into a program like Dreamweaver. Code such as `"` and `"` will have to be replaced (use Ctrl+F in Dreamweaver to replace the text) so that it becomes unformatted.



## PHP/MySQL RSS Feed Tutorial

---

```
<?php
header("Content-type: text/xml");
echo "<?xml version='1.0' encoding='UTF-8'?>";

// Set RSS version.
echo "
<rss version='2.0'> ";

// Start the XML.
echo "
<channel>
  <title>Feed title</title>
  <description>A description of the feed contents</description>
  <link>http://www.yoursite.com/</link>";

// Create a connection to your database.
require("db_connection.php");

// Query database and select the last 10 entries.
$data = mysql_query("SELECT * FROM table ORDER BY id DESC LIMIT 10");
while($row = mysql_fetch_array($data))
{
  // Convert database images data into actual image link.
  $row[Intro] = str_replace("images/", "http://www.yoursite.com.au/images/", $row[Intro]);

  // Continue with the 10 items to be included in the <item> section of the XML.
  echo "
  <item>
    <link>http://www.yoursite.com/article.php?id=".$row[id]."</link>
    <guid isPermaLink='true'>http://www.yoursite.com/article.php?id=".$row[id]."</guid>
    <title>".$row[Title]."</title>
    <description><![CDATA[\".$row[Intro].\"]></description>
    <comments>http://www.yoursite.com/article.php?id=".$row[id]."#Comments</comments>
  </item>";
}
// substr can limit the number of characters. First number is starting point while the second number is number of
// characters.
// otherwise just use $row[Intro].

// <guid> is an optional sub-element of <item> but recommended if you don't want to receive a warning when
// validating your RSS.

echo "
  </channel>
</rss>";
?>
```