Dynamic Data
Importing data from online resources: RSS

One way to add a fresh element your network-media projects is to use of the network to get up-to-date information from online sources. We will later discuss other techniques, but RSS is a popular one now. It is an XML-based way of formatting data. It is short for "really simple syndication" – which is another way of saying 'simple content distribution format.' The virtue of using XML is that many libraries can be written that will be able to read your RSS document because the language is explicitly defined. The 'magpie' project is a PHP library for reading RSS 'feeds' and converting them to some other format, like a web page. See for example:

   http://artcontext.org/edu/sva/webProg2/magpie/rss/ecoartivismo.php

This is a usage of the magpie library to display the three most recent entries in the ecoartivismo.net blog. Like almost all WordPress blogs, it has an RSS feed built in. We leverage that feature by using the URL:

   http://ecoartivismo.net/forums/?feed=rss2

You can see the source code for this feed by using 'view source' when looking at this URL (Firefox). A section of the document is reproduced on page two.
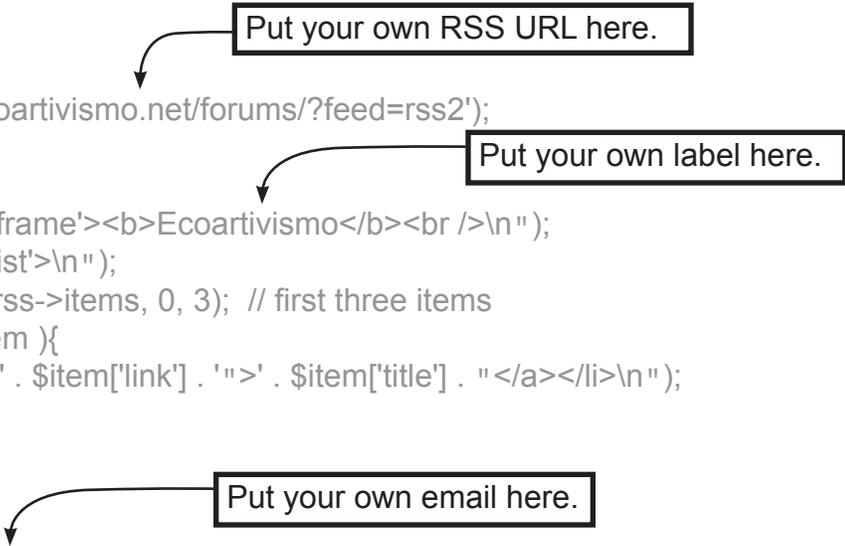
With the magpie API, we will use the following line of code to initialize the $rss object, which we will use to read all the entries found. But where is this fetch_rss() function? Because it's part of the

   $rss = fetch_rss('http://ecoartivismo.net/forums/?feed=rss2');

magpie library, we must include a library file at the top of our document like so:

```php
<?php
  include('rss_fetch.inc');
  $rss = fetch_rss('http://ecoartivismo.net/forums/?feed=rss2');

  if($rss) {
      echo("<div class='boxframe'><b>Ecoartivismo</b><br />\n");
      echo(" <ul class='boxlist'>\n");
      $items = array_slice($rss->items, 0, 3);  // first three items
      foreach ($items as $item ){
         echo(' <li><a href="' . $item['link'] . '">' . $item['title'] . "</a></li>\n");
      }
      echo(" </ul>\n");
      echo("</div>\n");
  }
  else {  mail("you@yourDOTcom","RSS","Debug the RSS script");  }
?>
```

Put your own RSS URL here.

Put your own label here.

Put your own email here.

You might ask, where does this 'rss_fetch.inc' file come from? Download the archive at the URL:
   http://magpierss.sourceforge.net/

The list of files and folders on the right reflects the files that are used when you include the rss_fetch.inc file. Just as we are using a library written by others, the authors of Magpie used a library called Snoopy.class.inc The 'cache/' folder is created and must have write permission for the web-server-user. (User that 'owns' the web server process.) It will contain feeds down-loaded from the net. That way, accesses to yourScript.php do not have to wait for the remote server every time yourScript.php is called. I have re-moved some inessential files from the archive downloaded from the sourceforge URL.

```
cache/
rss_fetch.inc
rss_utils.inc
yourScript.php
rss_cache.inc
rss_parse.inc
extlib/Snoopy.class.inc
```

```php
$items = array_slice($rss->items, 0, 3);  // cuts the $rss->items array down to a 3 element array
foreach ($items as $item ){                // loops and produces three $item associative arrays
    echo(' <li><a href="' . $item['link'] . '">' . $item['title'] . "</a></li>\n");  // echo variables from RSS
}
```

A useful debugging technique for arrays in PHP is to use the print_r() function, which takes the array as it's argument, e.g.; print_r($item);

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
        xmlns:content="http://purl.org/rss/1.0/modules/content/"
        xmlns:wfw="http://wellformedweb.org/CommentAPI/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:atom="http://www.w3.org/2005/Atom"
        xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
        xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
        >
<channel>
        <title>Ecoartivismo</title>
        <atom:link href="http://ecoartivismo.net/forum/?feed=rss2" rel="self" type="application/rss+xml" />
        <link>http://ecoartivismo.net/forum</link>
        <description>Arte y Activismo</description>
        <lastBuildDate>Sun, 24 Jan 2010 14:34:37 +0000</lastBuildDate>

        <generator>http://wordpress.org/?v=2.9.1</generator>
        <language>en</language>
        <sy:updatePeriod>hourly</sy:updatePeriod>
        <sy:updateFrequency>1</sy:updateFrequency>
        <item>
                <title>Animal Confusion</title>

                <link>http://ecoartivismo.net/forum/?p=438</link>
                <comments>http://ecoartivismo.net/forum/?p=438#comments</comments>
                <pubDate>Fri, 22 Jan 2010 22:58:14 +0000</pubDate>
                <dc:creator>dirtybomb</dc:creator>
                <category><![CDATA[Discussion]]></category>
                <category><![CDATA[extinction]]></category>
                <category><![CDATA[nature]]></category>
                <category><![CDATA[representation]]></category>
                <category><![CDATA[virtuality]]></category>

                <guid isPermaLink="false">http://ecoartivismo.net/forum/?p=438</guid>
                <description><![CDATA[The french publisher &#8220;Circonflexe&#8221; has published
a book named &#8220;Play of Images&#8221; that combines images of real wild animals with illustrations and
logotypes for Peugeot, Lacoste, Nestle and Sheba &#8212; all companies that use animals for their corporate
identities. This kind of children&#8217;s book contributes to the confusion between representations of the real,
on the [...]]]></description>
        </item>
    [... items omitted ... ]
</channel>
</rss>
```

sets up the general attributes of the feed before listing the specific content as item blocks

This is XML.... Not too pretty, but it has it's advantages, too.

You can include the listing of recent items in any PHP document, provided that you address the script correctly in your include statement, as for example:
<?php   include("rss/yourScript.php");   ?>