Creating a simple ad rotating system using ASP

by Manohar Kamath
August 9, 1999

Creating a advertisement rotator page is easy - ASP comes with an Adrotator component that's easy to use. The component, called AdRotator, is installed when you install ASP. The component has been a part of ASP installation since the first version of ASP.

There are 3 parts to the ad rotator system:

1. Ad rotator logic page
2. Ad images date file
3. Redirection page

Ad rotator logic page
Create a file say ads.inc, an include file that can be included on any ASP page on your site. This file will actually "create" the image links on the ASP pages. The page makes use of the file in item 2, the ad images data file to choose the image from the list of images.

The code is as shown below. I created a function getAd() so that this function can be called within any ASP page where you want the banners to show up. The advantage of putting the logic in the function is obvious - reuseability of code.

```
<SCRIPT LANGUAGE=VBScript RUNAT=SERVER>
Function getAd()
  Dim loAd
  ' Create an instance of the Adrotator component
  Set loAd = Server.CreateObject("MSWC.AdRotator")

  ' Set the target frame, if any. This is the frame the where ad
  ' URL will open up. If the HTML page does not find the target name
  ' the URL will be opened in a new window
  loAd.TargetFrame = "TARGET=new"

  ' Get a random avertisement from the text file
  ' This file is in the /ad folder
  getAd = loAd.GetAdvertisement("/ad/adrotator.txt")
End Function
</SCRIPT>
```

Ad images data file
create the ad information file, say adrotator.txt. This file has two parts

1. General information for all the images - this information is applied to all images that are rotated. This section contains 4 parts

   1. **REDIRECT** - the page that will handle the redirects. you can specify a absolute or a relative URL for this page
   2. **WIDTH** - Width of the images. If not specified, defaults to 60
   3. **HEIGHT** - Height of images. If not specified, defaults to 440
   4. **BORDER** of the image. If not specified, defaults to 1
2. Individual image information. This section is separated from the first by a single "*" on a separate line as shown in the code below. This section has four parts
   1. **Image source** - either absolute, relative or a URL like http://...
   2. **Redirect URL** - the URL to be sent when the user clicks on the image
   3. **Alternate text** - A brief description of the ad that appears in text browsers, or when the images are turned off
   4. **Ad weight** - a number between 0 and 4,294,967,295 that determines the frequency of that ad display. A weight of 0 means the ad will never be displayed. Higher the number, more frequent the display is.
      E.g. In the example below, the weights of two images are 2 and 3. So, the second image is 3 times more likely to appear for every 2 appearance of first.

Assuming the images are in the /images directory, a sample adrotator.txt file is shown below. The * separates the general data from the image info. Within the images info, leave a blank line between each image data.

```
REDIRECT adredir.asp
WIDTH 468
HEIGHT 60
BORDER 0
*
/images/fp2000.gif
http://www.microsoft.com/frontpage
Microsoft FrontPage 2000
2

/images/office2000.gif
http://www.microsoft.com/office
Office 2000
3
```

Redirection page
The redirection page, receives in its querystring, a variable named URL that contains the URL to redirect to, and another variable IMAGE that contains the image URL of the ad image. The IMAGE variable is more for statistical purposes, to keep count of what image appeared when and how many times etc. So, if you were to keep count of what images were clicked, you would log this into a database (there are better ways, but just trying to illustrate why the variable is passed).

A simple redirect page, adredir.asp, is as follows. It essentially retrieves the URL from the querystring and does a Response.Redirect to it.

```
<%@language=VBScript %>
<%
  ' Set the response buffer on
  Response.Buffer = True
  Dim lsURL
  ' Obtain the URL from the querystring
  lsURL = Request.QueryString("URL")
  ' Clear the response and redirect to URL
  Response.Clear()
  Response.Redirect(lsURL)
%>
```

Using the system
Include the file ads.inc on any ASP page you want to show the banner ad. Then, on the page, call the function getAd() at any point, however times you like to create as many ad banners. An example is as shown below:

```
<!--#include virtual="/ad/ads.inc"-->
<HTML>
..
<!--The ad banner goes here-->
<P><%=getAd()%></P>
...
</HTML>
```

Good luck!