How can a rotating banner image be done in Plone?

The first step is to create a Python script that will serve up the image you want. This article assumes that you want a random image from a folder to be chosen. Python-savvy readers can use more advanced techniques, such as making a list of images to load in the same order, etc.

Through the ZMI go to /plone_skins/custom folder, and add a Script (Python) object from the drop down menu. Give this script a name (this example uses “randimg”) and put this code into it:

```
from random import choice
# Import a standard function, and get the HTML request and response objects.
from Products.PythonScripts.standard import html_quote
request = container.REQUEST
RESPONSE = request.RESPONSE
RESPONSE.setHeader('Content-Type', 'image/jpeg')
# Get the current folder's contents, choose an image and return it
images = context.objectValues(['ATImage'])
return choice(images).data
```

Now that the script is in place, you can test it by pointing your browser to http://your.plone.site/images_folder/randimg. (Replace `images_folder` with whatever your random images folder is called.) It will work just like a normal image file on a web server. To add the image anywhere in your site, add the usual image HTML: **<img src=“/images_folder/randimg”>**. Every time the page is reloaded, a randomly-selected image from images_folder will be shown.

If you are using images other than JPEGs, you will need to change the `setHeader` call so its second parameter reflects the image type you’re using. For example, image/gif or image/png.

Also, the `objectValues` function retrieves only the file types you ask it to, in this case “ATImage”, which means images uploaded through the Plone interface. Images uploaded through the ZMI have a different file type, “Image”. This is an important distinction, so change the parameter if you need to. Note however, all content should be added through Plone and not through the ZMI. All content management should be done in Plone. Working with the ZMI in this fashion can result in unforseen problems. You have been warned. An alternative is to leave the parameter to `objectValues` blank, so that it returns all objects in the folder. However, if you use this option, make sure your random images folder has only images in it!

Update from Joel Burton on Plone.org

Using ‘objectValues’ will actually *retrieve* each image from the folder, even though you only want to use one. Not very efficient. It also is making no checks for things like effective/expiration date.

Better would be:

```
images = context.randomimages.getFolderContents({'portal_type':'Image'})
return choice(images).getObject().data
```

Overall, though, I’d recommend a different strategy: rather than having this script pretend to be the image, I’d have a script that returns a randomly-selected image **tag**. This will allow you to get a good image tag (w/height, width, alt text, etc.) from a randomly-chosen image. This would look like:

```
from random import choice
return choice(context.randomimages.getFolderContents({'portal_type':'Image'})).getObject().tag()
```

Then, rather than having code like `<img src="myscript" />`, you’d have your Page Template say `<img tal:replace="context/myscript" />`. This will give you a good image tag of a randomly-chosen image.

Revision #18
Created 2006-06-07 12:27:40 UTC by Rozett, Keith
Updated 2006-08-01 15:37:58 UTC by Rozett, Keith