Plone Widgets Client v0.1

Introduction

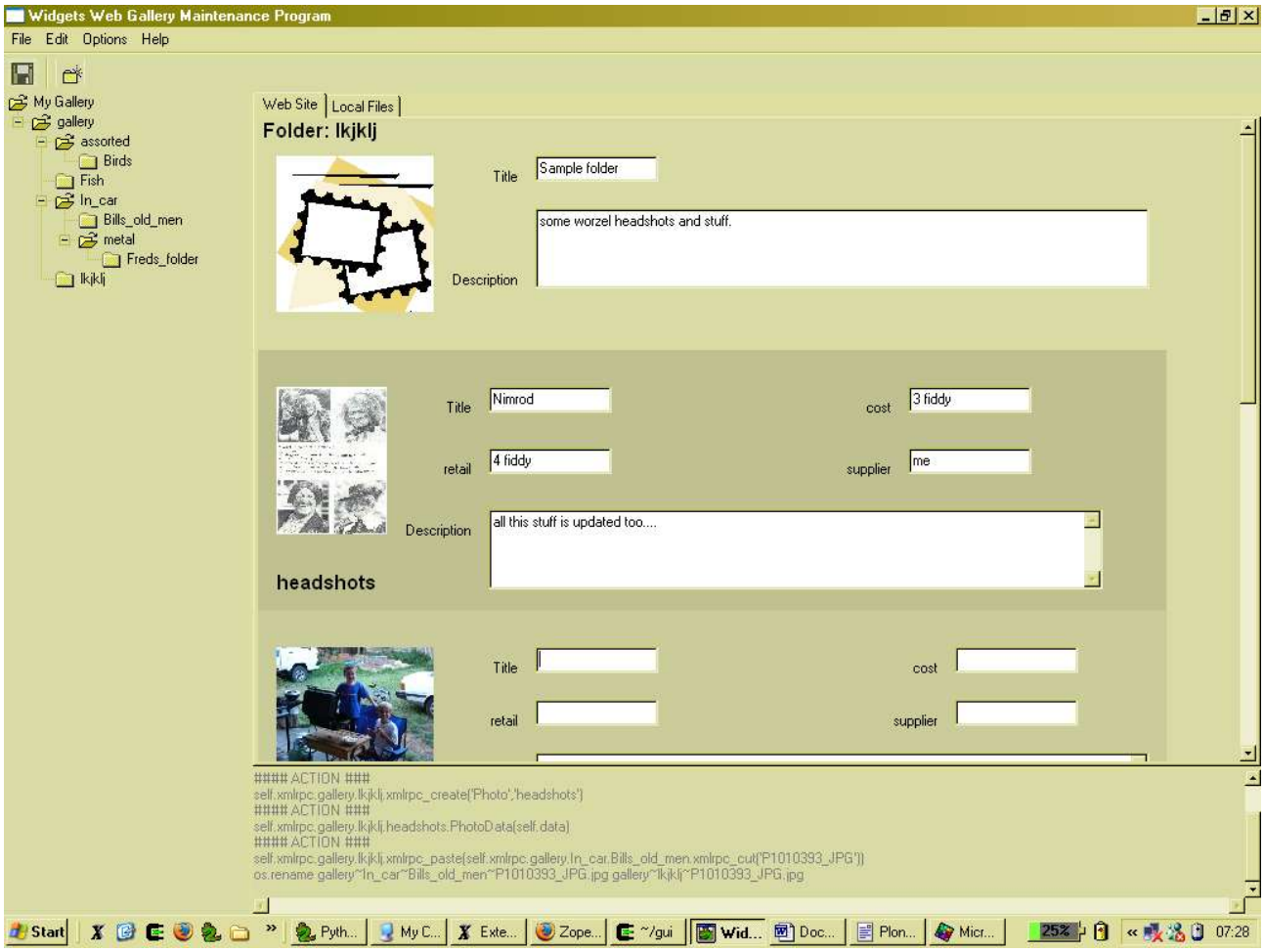
This software was created to make it quick and simple for a user to maintain a large web site. The goal is to soften the boundary between the internet and the PC. This program has been used in production: it started up inside 10 seconds and allowed fairly rapid site maintenance over a dial-up line. Partly this is achieved by batching the image-data shifting work and caching, so that most messages are text-only.

It was initially created for an art gallery: thus currently it supports CMFPhoto and CMFPhotoAlbum types on the zope side. The goal is to expand it so that it can support archetypes in general.

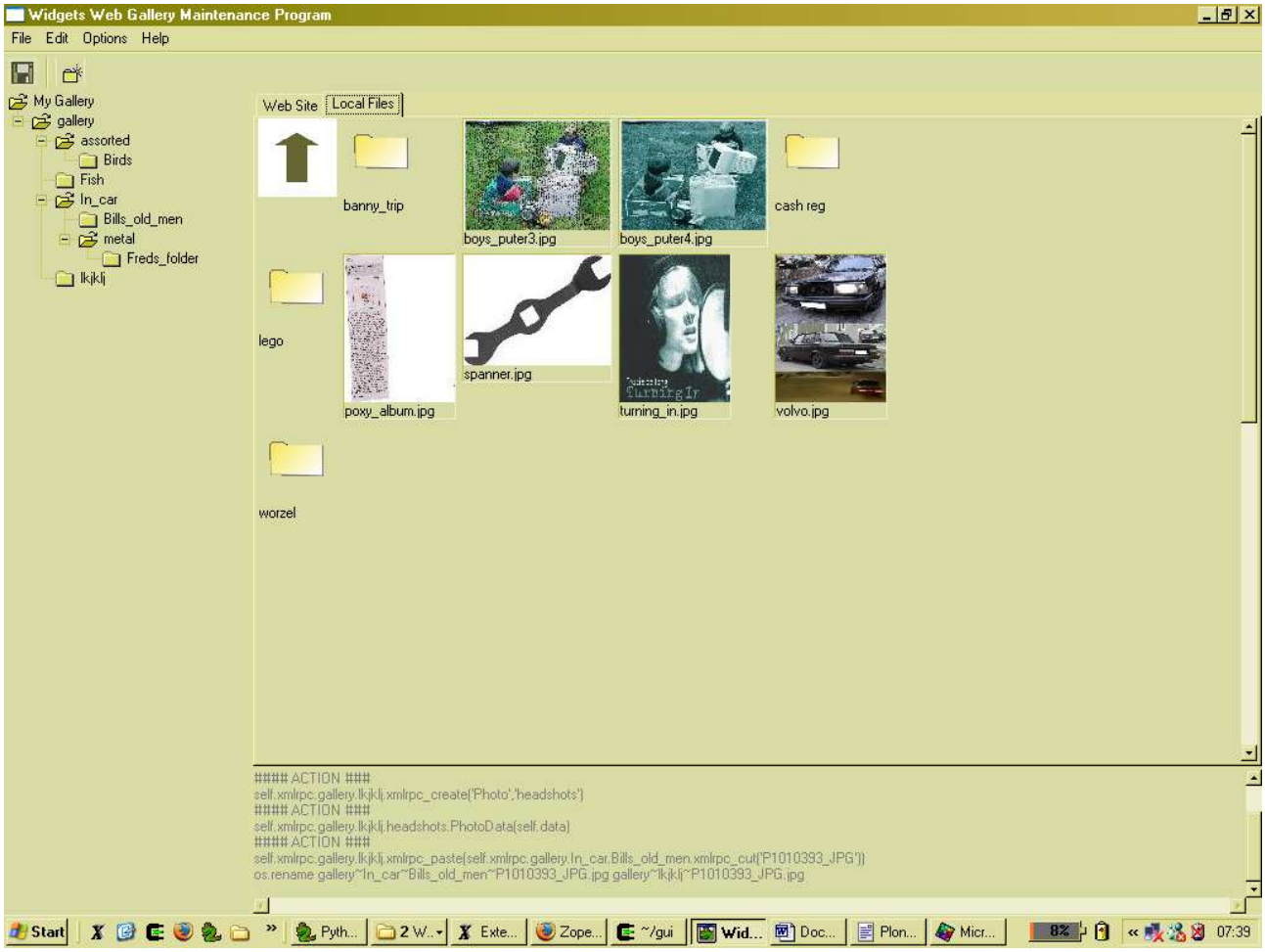
On the PC side it supports jpg files. The goal here is to expand it to understand a variety of file types, including Word and Excel. (Some Word/Excel modifications have been created in the past – but not included in this version. We also have Word/Excel Add-Ins that talk to Zope)

Here is a screen shot showing the program in action. This shows the Web Site side of things.

Site navigation is done using the tree on the left – contents of PhotoAlbums is shown on the right. The box in lower right is a log of the xmlrpc commands as they occur. Overtyping text fields and hitting save causes messages to zope updating the appropriate fields. New folders can be created using the File menu. Albums and photos can be moved using drag and drop (the nasty postage-stamps image beneath "Folder;lkjklj" is the handle for moving/deleting folders.



This screenshot shows the PC side of things:



On this side you'll see jpgs on your local filesystem. You can traverse the file system using the folder icons and the nasty up arrow. You can also drag and drop the images to the zope file system on the left. When you do the photo will be resized to the maximum size used in Plone, it will also give you the option of delaying the work of shifting photo data. So you can perform a lot of maintanence with very little latency, even over a dial-up line.

The parts.

This consists of a wxPython program and a modified Zope/Plone instance. Some of these modifications are zope methods that should be tidied into a product. The wxPython program has been built for Windows using the NSIS installer and py2exe.

Installing the Windows program

The installer creates a directory in "`C:\Program Files\`" called `WGallery` and creates a shortcut in the start menu. You have to manually alter your ini file to suit. The format for `WGallery.ini` is '`userid@url`' as shown in this example:

```
Rick@http://203.167.235.166/drawbridge/
```

When you start the program (Wgallery4.exe) for the first time you will be prompted for a local images directory (e.g. 'My Pictures'). That directory name will be written to the ini file and can be changed later in the options menu.

Setting up the Zope server.

This takes a little more effort! You will need to install the methods (external and internal) included in this package. You will also need a modified version of CMFPhoto and xmlrpclib.py.

Versions:
I doubt that much is particularly version dependent – but here are the versions I have been using:

Zope Version (unreleased version, python 2.3.3, win32)
Python Version 2.3.3 (#51, Dec 18 2003, 20:22:39) [MSC v.1200 32 bit (Intel)]
CMFPhoto (Installed product CMFPhoto (0.3))
CMFPhotoAlbum (Installed product CMFPhotoAlbum (0.3))
CMFPlone (Installed product CMFPlone (2.0-final))

I've run the zope versions across 4 different implementations, linux and Windows XP – but these are the versions running on my XP laptop now.

xmlrpclib.py:

This changes to allow the entry of userid and password via xmlrpc. This change was lifted from the zope site – credit to Amos for the clues. My copy of xmlrpclib.py in included with this release – my changes are marked in comments with 'kbm'.

CMFPhoto

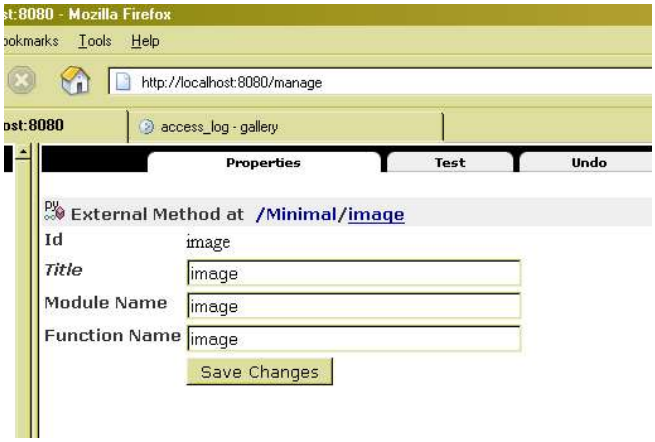
This is changed to make it pick up some field names on creation. It lifts the field details from photo_fields – you'll see this further down in the customs folder. I'm guessing there is now a nicer way to do this. Here is the change – again marked with my initials.

```
## kbm...
#  Add the fields describing this item
boffo = self.photo_fields()
for x in boffo:
   iobj.manage_addProperty(x,'undefined',boffo[x])
## kbm
```

This bit of code appears at the end of addPhoto() (and no, 'boffo' is not a good name for that field).

External Methods

Combining related external methods into a single python file is a Nice Thing to Do – and I didn't here. Nevertheless – there are some external methods. With one file per methods the name standard on the zope side becomes fairly simplistic – this example for images.py



Python Scripts (internal methods)

These are also required. I put mine in the custom folder. photo_display_sizes and photo_fields are there to let you define the max photo size (images are reduced to this size prior to upload) and names for the fields attached to the photo object (best not to change this right now – the windows end is hardcoded to accept those names)

Setup

At the plone root create a folder called access_log. In it create a page template called access_log. This is one of the dirtiest hacks in the system and I apologise for it now... its only there to support the blanket publish/unpublish steps that occur when you connect to your site.

Now create a CMFPhotoAlbum called 'gallery' in the root directory of you plone instance.

You should be good to go at this point. Best of luck and let me know how you fare. I expect a flood of constructive comments on how best to 'clean up' the messy bits.

Thanks

Kieran Basil Martin
Carterton, New Zealand.
kieran@widgets.net.nz