Taking Control of Windows

enfoldsystems

Outline

- About the application
- Python win32
  - How it works and how to use it
  - Some of the cool things
- Highs and Lows

enfoldsystems

On the way from OSCON

Windows GUI Programming

You don't hafta use C++,C# or Visual Basic here!

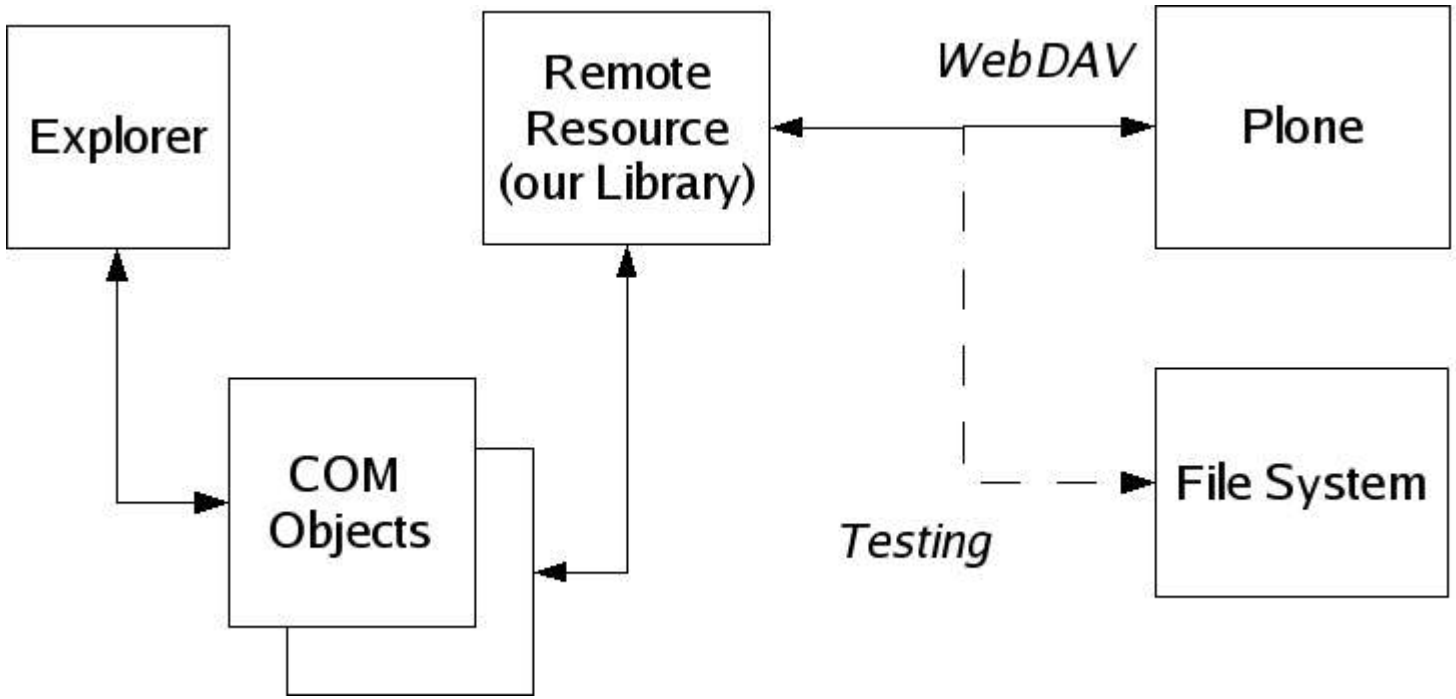
enfoldsystems

Concept

- We wanted to extend Plone into Microsoft Windows Explorer. But...
  - We didn't want to write C++
  - We have a whole bunch of Python expertise
  - So... enter win32 python bindings and
  - Write a whole bunch of win32 extensions so we could do whatever we wanted to Windows in Python

enfoldsystems

Project design



enfoldsystems

People

- Disclaimer: we had one person that is the Windows Python god
  - **Mark Hammond**
    - Lots of the stuff following is a direct result of his work.
    - Lots more got checked into the win32 project and is open source. This could not be done in Python before we did this project.
    - You may not be so lucky
    - So really most of this talk is by Mark Hammond, so complain to him...

enfoldsystems

For the impatient

- Demo?

enfoldsystems

So first

- We wrapped Plone's WebDAV calls so that
  - A call to Plone is sent via WebDAV is hooked and modified and sent back as a completer set of information
    - This set includes things like security, workflow and various other goodies
    - Unfortunately the Windows Explorer UI want's a lot of information

enfoldsystems

CMFPropertySets

- We wrote a tool called CMFPropertySets that allows
  - You to attach arbitrary properties to an object via WebDAV
  - This is nice because it can be plugged into any Plone instance
    - And also removed with no effect
    - This *is* being released as open source (when we get around to it)

enfoldsystems

Next...

- We wrapped these WebDAV calls into a filesystem like object so that we just access things with no knowledge
  - Of the source
  - Or the protocol
  - This allows us to maintain two implementations: WebDAV and FileSystems
  - This allows 200+ unit tests at this point

enfoldsystems

Then finally...

- We get to the Python win32 bit
- A project that wraps Windows calls in Python so they can be accessed by Python
  - http://sf.net/projects/pywin32
  - Easy to install executable
  - It wraps a whole bunch of C Windows API
  - The best book on this is “Programming Win32” by Mark Hammond

enfoldsystems

What is Python Win32

- A project that wraps Windows calls in Python so they can be called from Python
  - A bunch of modules that are importable from Python such as win32api, win32gui, win32shell and so on
  - Documentation comes with the install in the form of a windows help
  - I usually end up at the ASPN http://aspn.activestate.com

enfoldsystems

Our Problem

- We didnt actually have many of the win32 calls wrapped
  - So Mark wrapped them and added them to win32
    - IShellFolder, IShellView and so on...
    - IContextMenu is context menu handling
  - There's actually a few more to go:
    - IIconOverlay lets you overlay icons

enfoldsystems

So Windows Explorer...

- We register a COM server with Explorer that will respond to calls from Explorer
  - This is the scary bit (to me)
  - We create a COM object for every interface
    - So IShellFolder, IShellView
    - Can be 10,000 such objects
  - One note: this isn't a standard program, we are hooking into Windows
  - Explorer is the program, this is an extension

enfoldsystems

IShellFolderView

- Is called by Windows Explorer
  - Quick code diversion again
  - EnumObjects is called and this maps into ExplorerClient/EnumObjects

enfoldsystems

Property dialogs

- We borrowed some code from SpamBayes (again)
  - Written in Visual Studio and exported as .rc files
  - The parsed and loaded in Python
  - Translated into win32
  - Sounds complicated but... we can use a nice gui (VS) and then edit in Python

enfoldsystems

Property dialogs (2)

- There is a wrapper around win32gui which lets us
  - Take the .rc files, push them into win32 native widgets
  - You can then access the propertys and values from Python

enfoldsystems

Context Menus

- Context menu's are the menu's when you right click on something in Explorer
  - Actually this is pretty simple
  - You can make a win32 menu using the standard win32 code
    - Then you register this in the registry in the right point
    - Win32 will:
      - Call the relevant COM objects in the registry for the extension
      - Then it will try to invoke

enfoldsystems

Localisation

- Here we went fishing in Zope 3
  - And pulled out zope 3's i18n (along with interfaces)
  - Every time Plone Desktop is run any *new* i18n messages that are found are thrown into the catalog
  - All strings are localised through the catalog
  - This was actually a problem as we end up pulling way too much stuff
  - Currently have a Brazilian translation

enfoldsystems

Wrapping it up

- We used py2exe, which builds a nice big binary
  - Puts all the python code into a zip file of compiled code (so pretty hard to change)
  - Makes DLL's and puts them in the right place
  - Nice and self contained
  - ~2 MB download (of which thats mostly win32)

enfoldsystems

Highs

- Seeing it work
  - Was a big one :)
  - Manipulating Windows Explorer context menu's by changing things in Plone (eg Workflow)
  - Real pains:
    - Dragging and dropping was a real pain
    - Changing something would cause Explorer to refresh

enfoldsystems

Lows

- Testing
  - Isn't that easy. We set up a whole bunch of unit test so we could test
    - Explorer -> File System
    - WebDAV -> Plone
    - And bits in between but...
    - ...nothing to unit test Windows Explorer, any suggestions

enfoldsystems

Lows (2)

- Debugging
  - Not really an option. You can't run pdb in arbitrary Windows calls.
    - The flow of the program is from Explorer which is going to call you
    - Most of the time the problem is:
      - What call is being made by Windows and when
      - Or rather, what is **not** being made by Windows and when
      - Or rather, what call is being made by Windows and why does it have these arguments I didn't expect
      - Or rather !?@#!***

Lows (3)

- So we:
  - Although remote debugging via Wing/Komodo/Boa is possible (although not setup)
  - Lots of print statements
  - Register the .dll in debug mode
  - Go to *PythonWin* and open *Tools* > *Trace Collector Debubgging tool*
  - This will collect all print output

enfoldsystems

Lows (3)

- Reading the MSDN
  - To figure out what's going to occur when, this is really your only option
  - Fortunately Mark Hammond seems to read the MSDN in his sleep
  - Actually the MSDN is not that bad compared to some open source projects

enfoldsystems

Moving forward

- Lots more little neat API's we could wrap:
  - IconOverlay for example
  - Unit testing of the front end
  - COM Transformations are not done on the front end yet

enfoldsystems

Public API's in Python

- One selling point
  - You can now extend almost any aspect of Explorer in Python
  - Want a funky or weird column handler?
  - Want different property sheets?
  - All can be done in Python.

enfoldsystems

Conclusion

- Windows can be manipulated *a lot* from Python
- Mark Hammond is a god

- Any Questions?
- andy@enfoldsystems.com