



Contribution ID: 32

Type: **not specified**

## **Applications of python win32console module - a merging of character-mode applications and graphical user interfaces.**

*Tuesday 10 July 2007 10:00 (30 minutes)*

With the recent addition of the win32console module in python it is now possible to create GUI applications that have embedded console sessions. The win32console module is a wrapper around the Microsoft Windows console API which is the operating system component that enables character-mode applications to display data to a windows console. The most typical console mode application on Windows is the cmd.exe command line interpreter. By utilizing the capabilities of the win32console module, it is now possible to embed and enhance routine command line/shell interactions by the use of GUI applications. Examples of embedding cmd.exe sessions would be:

- wxPython based shell acting as a console replacement
- Divmod Nevow web pages using "server-push/comet" technologies as an interface around a remote console session
- cmd.exe session running in a Vim buffer

It is not only cmd.exe that can be embedded but any other character-mode applications such as a Cygwin shell or even the python or iPython interpreters.

The basic process flow is described as:

- Client process (1) such as wxPython or Vim editor starts an intermediary control process (2).
- The intermediary control process (2) is a child python process which imports win32console and allocates its own private console.
- This control process (2) then starts the target character-mode application (3) such as cmd.exe or python.exe.
- The control process (2) installs WinEventHook functions which are triggered whenever the target process (3) or any child processes of the target process write text to the console of the control process.
- The control process (2) traps the console output and then relays it the client process (1) by means of shared memory and event synchronization.
- The client process (1) can also generate command line input via shared memory to the control process (2) which then writes it into the input buffer of the target process (3).

From the perspective of the wrapped console process (cmd.exe or python.exe interpreter for example) there is nothing to distinguish its input as coming from an interprocess communication mechanism versus a person typing on a keyboard. Likewise there is nothing different in the output statements of the wrapped console process since it is just doing standard console output. The use of the win32console module wrapping the Windows console API simplifies interaction with the console processes since they will be running in their native command-line mode with no modifications required.

Technical discussion with slides and demonstrations will be presented.

### **Summary**

Methods used in, and demonstrations of the win32console python module to wrap and extend plain console applications on Windows operating system. Enables new applications for simpler use of console based applications or scripts; and enables easy remote execution of console sessions.

**Author:** Mr GRAZ, Michael (ITG London)

**Presenter:** Mr GRAZ, Michael (ITG London)

**Session Classification:** Python Language and Libraries

**Track Classification:** Python Language and Libraries