

Re: Is there an obvious way to do this in python?

Re: Is there an obvious way to do this in python?

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-08/msg01379.html>

- *From:* "Nick Vatamaniuc" <vatamane@xxxxxxxxx>
 - *Date:* 2 Aug 2006 12:36:44 -0700
-

HJ,

As someone already posted, the backend sounds very much like a database, so why not use a database: transactions, specific views for different users, limited access and so on = database!
Give PostgreSQL a try...

As far as presenting a different GUI to users, you can also do it based on the database. In other words have a common login screen and if the usertype from the database is returned as 'restricted' draw one interface, if it is returned as 'full' draw the full interface. Even if the restricted user will get the full interface up it won't be functional because the database would restrict writes to certain tables/columns.

Remote update of code is also possible, but you'll have to implement some kind of update server to which you can periodically send Python files, those files will be installed on the machine by the update server. You can try playing with Twisted to handle the networking. Or just write a simple script to send stuff over scp/ssh — that's what I would do (start the ssh server, install public keys and then just scp stuff over to the machines assuming they are online most of the time...).

The problem will be if something goes wrong in the updated file or with the update server then the whole system will be down (an off-by-one error in the GUI db client code and all of the sudden all your users will be writing bad data to the database... all at the same time). So you will need to do frequent backups of the database, but you probably know this already...

Hope this helps,
Nick Vatamaniuc

H J van Rooyen wrote:

Hi,

I want to write a small system that is transaction based.

Re: Is there an obvious way to do this in python?

Re: Is there an obvious way to do this in python?

I want to split the GUI front end data entry away from the file handling and record keeping.

Now it seems almost trivially easy using the sockets module to communicate between machines on the same LAN, so that I want to do the record keeping on one machine.

I want to keep the "server" machine as simple as possible – just doing record keeping on a stimulus response basis – I would prefer it to do one thing at a time to completion because this style of operation, though limited in performance, keeps a lot of hassles out of life – a transaction has either completed, or it has not – recovery scenarios are relatively easy...

Up to this point, I don't have a problem – my toy system can create a dummy transaction, and I can echo it from the "server" machine, with more than one "user" machine running – so I think it is feasible to have several tens of "data entry terminal" systems running, served by one not very strong machine.

Now what I would really like to do is to differentiate between the "User" machines, so that some can do a full range of transactions, and others a limited range.

And I would like to make this flexible, so that it becomes easy to introduce new transactions, without having to run around updating the code in all the user machines, with the concomitant version number hassles.

And I would like to do the whole thing in python – so my question is this – is it possible to do the equivalent of dynamic linking? – i.e. if I keep a list of what a user is allowed to do – can I somehow send him just the bits he needs to do the job, without having to change the static code on his machine? – it seems to me that the eval() thingy could possibly do this for me, by sending it data that makes it do import statements followed by calls to whatever... – will this work, or is there a better way?

Or has all this been done already? – and no I don't want a web server and php and browsers and Java and html or xml... – I want to write something that works simply and reliably – its just short message accounting type data...

– Hendrik