

Re: Memory problem – writing to database

Source: <http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2004-12/3054.html>

From: Bonj (a_at_b.com)

Date: 12/24/04

Date: Fri, 24 Dec 2004 13:35:03 -0000

> *Does not opening and closing db connection slow down the program ? Does it
> take a lot of resource to keep opening and closing the db connection ?*

Well... I'm suspecting that it won't slow it down as much as your current method will. Try it! One of the important principles in programming, is that the method that **seems** like it would be the fastest, best, or whatever, isn't always the fastest/best etc. at all – you just have to experiment with other ideas, and hey – I've given you another idea. You can take a backup of your current code and then chop and change it about as much as necessary – you don't lose anything by trying it out, even if the change is complete bladderdash! (though I suspect it's not...)

I'm not even sure that ADO classic doesn't have a clever form of connection pooling anyway, similar to ADO.NET. Even if it doesn't, it's even more probable that SQL server itself does. I also strongly suspect that by closing a connection, you are telling the RDBMS that it can perform necessary housekeeping code and can 'flush' things – your current idea of keeping a recordset (and hence also a connection) open all the time you're starving it of the opportunity to do these things, but you're still inserting loads of data.

Don't worry about the fact that you're constantly opening new connections – instead, **try it**. You might be impressed with the results. If you still notice the same slowdown, **then** post back and tell me it's a crap idea.

And I don't know whether you are, but if you are using task manager to measure the memory usage of the program, please find something more accurate if you want to report credible results of serious testing. You could use process explorer from sysinternals, or alternatively an internal method, such as having a column in the database table logging the time the record actually made it to the database, and a column where you insert the time that the record was actually received (which you remember as soon as the record is received) – although if you use this method don't let the server calculate one of the times and the client calculate the other – have the client calculate both. The difference between the times then gives you a rough indication of the overall latency.

Use an accurate performance testing method (more accurate than looking at the memory usage in task manager) on your current method, and the same test on the method I suggest, implemented exactly as I have described, and post back the results.