

## Re: CDatabase, CDaoDatabase

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2004-04/1342.html>

---

*From:* Tim (*Tim\_at\_NoSpam*)

*Date:* 04/15/04

Date: Fri, 16 Apr 2004 10:16:57 +1200

Hi,

A generalised statement on the order of performance:

DAO (obsolete),  
raw ODBC calls  
CDatabase / CRecordset – minimally / insignificantly slower than raw ODBC in  
production.

DAO will outperform ODBC (and so CDatabase) considerably in single user situations or where there are few users. DAO is for MS Access databases only. So the only appropriate comparison here is DAO to an access DB vs. ODBC to the same access DB. You "can" use DAO via MS Access to access EG SQL Server via linked tables, but that is like posting a letter to your neighbour and having it routed via Vladivostok. ADO is no more efficient than ODBC currently – I have no doubt that it will be in the future and is the technology for all new projects.

Simply put, your comments do not tally with experience.

SetQueryTimeout(0) will *\*not\** have any impact on performance – it merely sets the maximum elapsed time that a query can run before it times out and is automatically cancelled (for being a resource hog). Setting it to Zero indicates that the query can take as much time as it wants. If you observed a difference then it will be for some other reason.

There is considerably more to database performance than making quick assessments based on some MFC code. You have to start with a properly designed database, appropriate indexes, use the most suitable data access technology (IE CDatabase vs ADO vs OLE DB) from a coding perspective, and then properly use the classes as intended. I suggest that if you want best performance and you are using SQL Server, then get to know the SQL Server Profiler since by far most elapsed time in a database system will be spent running queries and that will give an indication of 'cost' of each query.

If you have a specific problem, please define it and post a request and someone is bound to assist.

Your differences could be due to a combination of factors including: using a debug build, low memory, concurrent processes and IO, Exclusive vs. Shared database access, record locking and so on.

I suggest that you check you made your comparisons on an equal footing: use a Release build and ensure you are comparing apples with apples. At the end of the day, do not expect a database centric application to run any faster under C++ than VB – since it is database centric. The CPU time used may be less with C++, but otherwise expect comparable performance. In some circumstances and with careful optimisation, a C++ database project can thrash the daylights out of a VB equivalent, however the situations where this degree of effort are warranted are rare.

– Tim

"TomTom" <sergiogiogio@yahoo.fr> wrote in message  
news:d5c8958d.0404151026.71de9cca@posting.google.com...

> Hello,

>

> *I found that CDaoDatabase and CDatabase to be incredibly slow compared  
> to native ODBC calls... An UPDATE, DELETE, INSERT is typically 5 times  
> slower, and for a SELECT, it is almost 10 times slower!! The C++  
> project is actually even slower in this respect than the VB project!!*

>

> *Browsing a bit into MFC code, I found that setting SetQueryTimeout(0)  
> partially solves the issue, the queries run much faster.*

>

> *I am very surprised I found nowhere in microsoft documentation any  
> mention of this, and, more surprisingly, very few posts about the  
> slowness of this MFC component.*

>

> *Any comments/thought about this? Do you know any other tricky way to  
> speed things up even more?*

>

> *thanks in advance,*

>

> *SerGioGio*