

Re: TADO vs. TQuery

Source:

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2005-06/msg00246.html>

- *From:* "Arnie" <none>
 - *Date:* Wed, 22 Jun 2005 16:04:39 -0400
-

"Maurice Telkamp" <maurice.telkamp_at_beeone.nl> wrote in message [news:42b7c5c9\\$1@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:42b7c5c9$1@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)
> "Allan Nielsen" <ace-allanREMOVE@xxxxxxxxxxxxxxxx> wrote in
> message news:42b7ab33@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>> How much faster is ADO typically over a simple TQuery using
>> odbc?
>> I have an application that uses ODBC to connect to MS SQL
>> Server 2K. Will
>> this benefit greatly if I changed to ADO?
>>
>> Thank you.
>>
>> Allan
>>
>
> I guess most of the guys here will claim that BDE is pretty
> fast, but I have no complaints about ADO. But I would not
> consider speed the dealbreaker here. Go for ADO, you will not
> regret it. Ease of use, ease of configuration, capabilities
> etc. Besides that, using ADO, you can still use ODBC (although
> I would not recommend that)
>
> Maurice
>

It will take a while to 'tune up' ADO objects to get BDE speed.
But if you're leaving BDE (highly recommended) it's worth the
effort. And for God's sake, don't use Table objects.

Also, don't knock ODBC.

Here's a little story. We used BDE several years ago. We were
using an Oracle DB. Eventually we started using DOA (Direct
Oracle Access) for the heavy duty console apps. DOA is written
using Oracle's OCI and was much faster than BDE. Then we had to
support DB2 and eventually SQL Server. I wrote two sets of
database objects. One (RMS) that provided the same interface as
BDE but used TADO under the covers. The other (GDB) provided the

Re: TADO vs. TQuery

same API as DOA. It used either the DOA DLL for Oracle or the RMS DLL for DB2 and SQL Server. Either was dynamically loaded depending upon which DB was in use. DOA was still MUCH faster than using ADO. Life was good – well, OK anyway.

Recently 'we' decided to port those heavy duty console apps to VC++. In the end, I ripped the guts (ADO) out of the RMS DB objects and replaced them with ODBC. I also got rid of the GDB/DOA objects. One reason for this is that we're going to 64 bit real soon and DOA is based on the VCL (TDataset). So, now I had the RMS DB objects, providing a BDE-like API, using ODBC. This made for a pretty easy port. Porting away from the GDB/DOA objects took more work but was largely mechanical. Note that we have around 140 of these console apps. Most used a combination of the RMS and GDB obje