

Wednesday, March 1. 2006

ADODB vs ADODB-Lite? It's not a war...

Making a short (yeah, like that'll ever happen



) commentary on iamsure's post about ADODB over on

<http://www.kabal-invasion.com/bugjuice/>. Iamsure is a developer on the [Blacknova Traders](#) space strategy game.

First up ADODB is a fantastic library. But it's not perfect. I'll be murdered for saying that perhaps - but it's not. It has problems - now these are problems unlikely to be noticed in the right circumstances. An opcode cache running on a dedicated server can hide a few warts



In short - it uses a lot of memory, it's ungainly, it's slow (usual limits apply - see second paragraph), it had a stupid security flaw until v4.71 and it just feels plain bloated. As a player of TL said after the recent switch to ADODB-Lite: "Hey, did you upgrade the server or summit?" Under stress ADODB-Lite performs significantly better. It still performs better than ADODB even with an opcode cache installed.

It's a matter of personal circumstances. ADODB is great - but it sucks a little too much RAM for something hosted on a shared host or a cheap dedicated server seeing high usage. ADODB-Lite is Lite. It doesn't suck a lot of RAM, and it's faster. So nah nah nah..



Feature wise the news is hardly grim. ADODB-Lite supports bind variables and has done for a long time - it's the basis of Partholans Data Access layer where every SQL query is bound. It supports transactions via an optional module - i.e. you need only include it when required. In fact it supports a bit with these modules. And more can be added if someone feels the need. That includes the PEAR and DATE modules to name a few popular ones. It also supports ADODB's XMLSchema which is used in QS-Evolved's installer, and also in our UserModules.

It does not (to date) support resultset cache-to-file or performance monitoring. The former I don't really mind (I could work around it) and I'm looking into whether a perfmon module is possible in any case - hopefully I can figure that out in the next few weeks.

The latter is difficult to assess. On the one hand many DBMS's have some form of intelligent memory cache, on the other it may be badly configured or disabled - typically by setting a low or zero cache size. In those cases a file cache may boost performance, assuming file access is faster than database access (varies). In any case ADODB does not offer an "intelligent" cache. It's not a simple decision to make anymore. MySQL 4.0 is standard fare these days.

An interesting mention was PostgreSQL string escaping. I reported this months ago incidentally to John Lim - even blogged it for what it was worth. Nothing was ever done about it despite multiple mentions, and I was migrating to ADODB-Lite in any case so I dropped it. The fix was in fact to uncomment the qstr() function in the postgres64.inc.php driver. Stupidly easy - never done.

It took a much more visible public security notice (<http://nvd.nist.gov/nvd.cfm?cvename=CVE-2006-0410>, and also a Gentoo Linux security mention) to get it finally resolved. Bad publicity has that effect. I can't say I was entirely happy using a library claiming to escape data when it still did nothing of the sort months after its

reported as a security flaw in a less public manner.

ADODB-Lite had this fixed within a short time of my reporting it. The security conscious at work...

Posted by Pádraic Brady in PHP General at 18:39