

Re: Enterprise Library – DataAccessApplicationBlock – MySql/DB2

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2006-03/msg00410.html>

- *From:* David Thielen <thielen@xxxxxxxxxxxxxx>
 - *Date:* Mon, 13 Mar 2006 10:46:27 -0800
-

Hi;

Good points and I agree. JDBC does do this a lot better, but they have had a lot longer time to get it to that point. I do think the one major thing left to address in ADO.NET is a standard way of setting the parameters in the SQL statements. I don't think there is any inherent reason why setting parameters in a select must be different for each database.

I think it is going to take the next version of ADO.NET (2.1?) to get us to where I would like it to be. Where for simple CRUD the only difference is the connection string.

—

thanks – dave
david_at_windward_dot_net
<http://www.windwardreports.com>

"W.G. Ryan – MVP" wrote:

David:

The ApplicationBlock isn't ADO so it's portability limitations don't reflect ADO in any way.

The fact that there are specific providers and code handled differently is definitely by design as opposed to them leaving something out. In order to take advantage of many features of Sql Server or Oracle, you simply can't use generic features. The more generic you go, the more you compromise on either functionality or performance, and to the best of my knowledge, this tradeoff is inherent and really couldn't change. You can write agnostic code but that comes at the expense of functionality or performance, something most developers don't typically want to compromise on. So you can have it either way, going totally portable or totally specific and the current arrangement provides the provider specific stuff by default. You

Re: Enterprise Library – DataAccessApplicationBlock – MySql/DB2

have to write your own generic features.

the DataSet/dataTable model is truly portable though, it couldn't care less where data came from or where it's going. You decide how to get it where you want it to go. This separation seems logical and solid IMHO.

As far as 'true' portability. You can scrap Paramaters altogether (not advised but possible), and use dynamic sql. SQL is ANSI standard so the code should fire against any Sql Compliant DB. You can use the ProviderFactory to create the commands and connections and adapters. Yes, there's a lot of problems with this, performance aand security being two of the bigger ones. But everything is a tradeoff...

"David Thielen" <thielen@xxxxxxxxxxxxxx> wrote in message
news:1DEF5911-6959-45C1-BB97-0D2DD52EB93A@xxxxxxxxxxxxxx

Ok – thanks this lets me know to not use DataAccessApplicationBlock – you just saved me some time.

It looks like I have to write my own code to handle each client. Maybe in 3.0 ADO will be like JDBC where it is truly portable...

—

thanks – dave
david_at_windward_dot_net
<http://www.windwardreports.com>

"Luke Zhang [MSFT]" wrote:

The DataAccessApplicationBlock currently only includes classes that are specific to Microsoft SQL Server and Oracle. You have to use ADO.NET with DB2 and/or MySql directly.

Regards,

Luke Zhang
(This posting is provided "AS IS", with no warranties, and confers no rights.)

