

Re: Ent Library Application blocks

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2007-08/msg00059.html>

- *From:* "Kevin Spencer" <unclechutney@xxxxxxxxxxxxx>
 - *Date:* Wed, 1 Aug 2007 08:17:02 -0400
-

Hi Gary,

Whether you're missing something or not is really determined by what your needs are as a developer. In my own experience over a dozen years, I have discovered that my scenarios have changed quite frequently, and I have needed different functionality from databases from one project to another. If you work on and maintain only one project, and the data requirements of that one project are well-defined and unlikely to change, Data Access Application Blocks might be more than you need. But if you have changing requirements over a long period of time, and a changing set of team members over time, they make good sense.

The idea of Data Access Application Blocks is derived from the same process that gave rise to assembler, high-level programming languages, functions, structures, and object-oriented programming. That is, certain types of operations require the same or similar sequences of instructions and/or data to be performed. So, rather than writing redundant code with many possible points of failure, similar types of operations are combined and encapsulated for re-use. As my Uncle Chutney sez, "Big things are made up of lots of little things." If you enjoy typing the same thing over and over again, and having more code to maintain, you certainly don't need any of these things. OTOH, if you want to have a smaller code base to work with, less code to debug and maintain, and fewer points of failure, encapsulation of common functionality is the best way to go.

For example, as you've already mentioned, most database operations require a couple of common things: A Connection, A Command, and a Container for any results. Sometimes the Container is unnecessary, such as when performing INSERT or UPDATE operations, but at the very least you want to have some kind of return value to indicate status, success, or failure.

Of course, there are many different sorts of databases and other sources of stored data. This means that these objects must be adaptable to different database and data source types. But they do have a number of things in common. So, the .Net Platform has the System.Data namespace which contains a number of base classes that can be derived from to create data-source-specific classes.

Re: Ent Library Application blocks

Still, within a given set of database parameters, to perform an operation of some kind, you're likely to do at least the following operations for each:

1. Open a Connection
2. Create a Command
3. Execute the Command
4. Close the Connection

Now, that's 4 operations common to all database operations. The details vary, but the basics remain the same. Now, why rewrite the code that does these things each time you perform a database operation? Why not encapsulate them into a single operation, or perhaps 2?

Obviously, if doable, that is a desirable scenario. For one thing, rather than having 4 X (however many times your code needs to work with a database) operations to debug in your code, you only have 4 X (1 Method call) to debug and maintain. One point of failure, and one Method to maintain.

Now, the complexity arises out of those details that differ from one database operation to another. The Connection String, for example, may differ. The type of Command may differ. The result may or may not include data. And so on. That is why the DAAB are designed as they are. There are lower-level encapsulations of commonly-used items like Connections, and higher-level implementations of combinations to handle recurring common scenarios.

Of course, you may not need something quite as all-encompassing as the Microsoft Data Access Application Blocks. You may want to implement something more specific and light-weight for your own needs. And the .Net platform has all the pieces you need to do this. For example, my projects almost all use SQL Server. So, I can create (and have created) similar, simplified Data Access Application Blocks of my own for my sorts of projects.

--

HTH,

Kevin Spencer
Microsoft MVP

Printing Components, Email Components,
FTP Client Classes, Enhanced Data Controls, much more.
DSI PrintManager, Miradyne Component Libraries:
<http://www.miradyne.net>

"GaryDean" <GaryDean@xxxxxxxxxxxxxxxxxxxx> wrote in message
<news:ebc5n190HHA.4824@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

I have just been through the docs on the Data Access Application blocks and it seems that they complicate things more than make things simple. To me it seems that there is nothing more simple and straight forward than

Re: Ent Library Application blocks

writing simple stored procedures and executing them from .net code using easy to understand connection strings.

I'm looking for opinions here from those that have used these tools. Am I missing something?

--

Regards,
Gary Blakely