

## Re: More ASP.Net Newbie Questions

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-11/6547.html>

---

**From:** Dan Bass (*danielbass*)

**Date:** 11/29/04

Date: Mon, 29 Nov 2004 16:15:52 -0000

Ah, another factor of great significance. If you're not using VS.Net it'll make things a lot less obvious...

I'm not sure "break ... into" is the right way of looking at it. Because everything is stored in objects, they've been grouped in relevant classes, which brings us ADO.Net. After understanding how to, for example, populate a grid from a stored procedure, it clicked into place as to why things are the way they are. A connection object, for SQL Server for example, can either be from a SqlClient, Odbc or OleDb library.

The first keeps all code talking to Sql Server in .Net managed code, making it faster, more efficient and less prone to leaks. The Odbc / OleDb have been provided to allow to talk to any database, but this does come with having to use the drivers for the database which will lie in native (unsafe) code which is slower, and have been known in the past to cause trouble. There's also an OracleClient which works well too with Oracle databases.

The Command is then what you're doing with this connection, whether calling some SQL, a stored proc etc...

Finally, readers allow you to parse the record set you've got back if you need too.

It's easy to create datasets from commands, and these can in turn be bound to most Asp.Net controls.

The first thing I do in a new project is decide the most common way data binding will take place for datagrids, for example, then write up a static global method called "BindDataGrid" which takes in a connection, command and control and binds it. From here on in, all I need to do is simply call my method with a few parameters, and there's a populated datagrid. Easy!

In regards to your final point, making grids and controls in general do different cosmetic things is simply a case of creating an event handler for the "Item\_Created" event, then checking the condition and deciding how this changes the look of that cell or row... again, once you learn it, it's easy and only two or three lines.

It's going to be a lot harder developing ASP.Net without Visual Studio .Net although it seems there is support for .Net in dreamweaver '04.

"MDW" <MDW@discussions.microsoft.com> wrote in message news:F09B2971-6EEF-4778-AF40-428EAEDDA137@microsoft.com...  
>> 2. *The debugging is 100x's better too, with the ability to step through web sites (even on remote servers), watch variables, etc etc...*  
>  
> *I will grant you that – I found that the error handling and error messages were a lot more informative w/ ASP.Net.*  
>  
> *Unfortunately, my boss has the strange fascination w/ Macromedia, so I'm developing in MX Dreamweaver 2004. It's a competent enough program, but since we have neither Visual Studio.Net nor VB.Net installed, any of the benefits granted therein are unavailable to me.*  
>  
> *I really really wish someone could explain why they had to break the Connection and Recordset objects into, like, 37 different things. Command object? OLEDataReader? That's really what caused me to re-examine .Net.*  
>  
> *Seems to me like you cede a lot of control to the compiler, especially for doing things like outputting a recordset to the screen. Sure, it takes less code to bind it to a datagrid, but don't you lose a lot of fine tuning you get by stepping through the Recordset? For instance, say that if the value of the first field was less than 0, change the background color of that row in the table to red. In classic ASP, that's very easy to do and there are probably a million different ways to accomplish that. But when you bind data to some sort of html control, isn't it pretty much an all-or-nothing deal?*