

Re: Data Environment – any pitfalls?

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-06/msg01347.html>

- *From:* "Ralph" <nt_consulting64@xxxxxxxxxx>
 - *Date:* Tue, 20 Jun 2006 07:48:29 -0500
-

"MyndPhlyp" <nobody@xxxxxxxxxxxxxxxx> wrote in message news:ufAraC8kGHA.4508@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Newbie (mostly) to the Data Environment object.

The short story is that I like it. Adding fields to forms and reports is just a matter of drag'n'drop. In the past I've been doing things longhand (i.e., ADODB.Connection, ADODB.Recordset, etc.) but this really changed my outlook. Right now I'm coding on Win2K with VB6 but I expect to see application running on Win98–WinXP (and who knows what else).

I'm about to commit to coding using the Data Environment throughout a

rather

complex application and I'd like to know from you gurus...

Are there any significant pitfalls and/or cautions I should be aware of regarding the use of Data Environment before I wholly embrace this idea?

To amplify a bit on the other's suggestions... (tho I am certainly no guru)

I appreciate the usefulness of the DE. The DE was introduced with ADO 2.0 as a 'wrapper' to provide a combination of easy-to-use graphical designers, code generators, hierarchical data, 'stored queries', and centralized event and error-handling – as a direct response to other popular tools such as PowerBuilder.

It can dramatically improve development times. Unfortunately it is a wrapper and thus doesn't expose all the features of ADO and there can be an impact on performance due simply because of the secondary level of indirection. Also it represents "Version 1" and MS never advanced the tool. (The same is true of Data Binding – a separate but closely related technology.)

It is useful to learn the DE not only because the built-in tools simplify

Re: Data Environment – any pitfalls?

learning, but it introduces programmers to a useful model for data aware classes, event, and error management, and reuseability. A model that can be easily mimicked to good effect when using ADO directly later. Too often beginning programmers jump straight into ADO programming and end up writing disjointed, data-centric code, monolithic data forms, and never become aware of the many features that ADO and Data objects can supply to improve the resulting product.

I use the DE and Data Wizards extensively in prototyping and as part of an agile approach to development at large. Especially as part of an in-house suite. The speed of development is invaluable in creating proof-of-concept and place-holder (test-bed) objects. In many cases, it serves quite well and survives into the finished product.

–ralph