Subject: Re: Structure field concatenation
Posted by Ben Tupper on Thu, 31 Aug 2000 20:52:03 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

>
> I meant computationally expensive. For some reason putting and
> extracting large data arrays in structures is not very efficient.
>

Hello and thanks to all,

My experience has been quite the same, shuffling large data arrays in and out of structures is time consuming.

I guess I'm headed toward pointers as fields. In truth, that is what I have now despite the example I posted, but I can't lose my original data set (embedded in a database object), so I shall have to truck around a copy of the user's last query definition in the form of a resulting data structure. The issue comes to a head when the user calls for the data structure (via GetProperty method.) I'll be standing there with a pointer filled structure (correctly queried with something foolish like 'give me the sum of the males and the transitionals'). I could just pass the data structure full of pointers and allow the user to dereference the field pointers, or I could build an anonymous structure filled with the derefenced fields. The former is easy, but relys upon the good judgement of the caller not to free the pointers. The latter seems safer, but I will still need to build a structure.

Bing! I think I just got it. Make each of the structures (original and queried subset) the same 'automatically defined' named variety complete with pointers for each field. I'll still need to build a new derefenced anonymous structure when the caller demands the queried subset data (or even all the data for that matter) but I can live with that.

Thank you! Thank you! Thank you!

Ben

--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net