

Re: ping DelM: colnitalize

Source:

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2005-09/msg00190.html>

- *From:* "Del M" <Del.Murray@xxxxxxxxxxxxxxxx>
 - *Date:* Wed, 21 Sep 2005 08:57:16 -0400
-

Guillem,

No, I have not. It appears that in Delphi 5, when an Isapi is instantiated that if an ADO connection object exists, there is some strange behavior. During the creation of the connection object, which is not in my code but rather in the ADO unit that is in the "uses" clause, if cointialize has not been called, then the object cant be created. This only happens under certain variations of Windows Server 2K and some NT4 Server with certain hot fixes applied. I have a web farm and this only occurs on one machine in that web farm. If I include the "coinit" unit as the first unit in the uses clause of the main dll, then the error goes away because that program is run before the datamodule is created. I believe that Kevin Frevert passed that around some 1 or 2 years ago and it has been fine. Using that unit, I have never had to actively call coinit or couninit in any of my ISAPI programs. I do not start separate threads in my ISAPI, they are in effect 1 thread each. I saw some obscure references in some M\$ documentation that stated "cointialize is required inorder to create an ADO object", leaving me with the impression (which might be wrong), that coinit has nothing to do with ADO execution, only with creation of objects that use ADO. THat would properly explain the behavior that I see where a connection object on a webmodule will cause the program to abort when the object is created and coinit is not called. This happens even if there is no connection string and no open, it is just during the create process.

Lots of people have said to call coinit before each execution. I think that is not required, otherwise, they should have written the ADO unit to do it for us. I can not imagine changing all my code to call coinit just before an "myDataObject.open" on a stored procedure that returns data or a tadodataset that returns a result set. Coinit is supposed to be so that ado can talk to oledb library. If my tadodataset referes to a connection object, then I think that the connection object is the actual pipeline to the oledb call and that is where the coinit is required.

So I then put CoUnint followed by CoInit in my "beforedispatch" event, ... didn't make any difference.

I am switching to Delphi2005 over the next couple of weeks and hope that the problem will go away.

What really pisses me off is this ... if "coinit" is so damn sacred, why

Re: ping DelM: colInitialize

doesn't it happen automatically. We invented the starter for the automobile about 100 years ago and don't have to get out of the car and crank it by hand every time we want to start it .. it seems to me that those guys at MS that claim to be so brilliant, could have applied the same principles to their software and made the coinits built in .. computers are supposed to do the work, not me.

I know I'm ranting ... I'm just tired of being the victim of "unannounced" features in all this stuff. It's as bad as the old days when you bought dBase and found out the next release was the one that really allowed you to write your own procedures despite what the "box" said in the advertisement.

I have been in this business since the first IBM 360-30 hit the street and on mainframes I never, ever had these kinds of problems.

• *Follow-Ups:*

- ◆ *Re: ping DelM: colInitialize*
◇ *From: Kevin Frevert*
- ◆ *Re: ping DelM: colInitialize*
◇ *From: Vitali Kalinin*

• *References:*

- ◆ *ping DelM: colInitialize*
◇ *From: Guillem*
- Prev by Date: *Re: ping DelM: colInitialize*
- Next by Date: *ADO Row Marked for Deletion*
- Previous by thread: *ping DelM: colInitialize*
- Next by thread: *Re: ping DelM: colInitialize*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*