

## Re: ODBC/OLE DB Connection Pool

**Source:** <http://www.tech-archive.net/Archive/Data/microsoft.public.data.oledb/2004-11/0146.html>

---

**From:** Bob Barrows [MVP] (*reb01501\_at\_NOyahoo.SPAMcom*)

**Date:** 11/26/04

Date: Fri, 26 Nov 2004 13:11:47 -0500

Erland Sommarskog wrote:

> *Bob Barrows [MVP] (reb01501@NOyahoo.SPAMcom) writes:*  
>> *If you read Stuart's post carefully, you will see that you do not*  
>> *have to do anything to turn Session pooling on. It's on by default.*  
>> *You have to do extra work to turn it off.*  
>  
> *Well, in VB it as easy as:*  
>  
> *Public Function Button1\_Click () As*  
> *Dim cnn As new ADODB.Connection*  
> ...  
> *Set cnn = Nothing*  
> *End Function*  
>  
> *And if there is no global ADODB.Connection, there will be no pooling.*  
> *(Which Stuart also discussed in his post.)*  
>  
> *If the above is possible to do in ASP, I don't know.*

Oh wait, It HAS been a while since I read this article. I now see this paragraph:

\*\*\*\*\*

### Tips for ADO Users

The ADO Connection object implicitly uses IDataInitialize. However, this means your application needs to keep at least one instance of a Connection object instantiated for each unique user—at all times. Otherwise, the pool will be destroyed when the last Connection object for that string is closed. (The exception to this is if you use Microsoft Transaction Server. In this case, the pool is destroyed only if all of the connections in the pool have been closed by client applications and are allowed to time out.)

Note If you open a Connection object, remember to close it. Do not rely on garbage collection to implicitly close your connections.

\*\*\*\*\*

I do question whether this means that the pool won't be recreated the next time a connection is opened

microsoft.public.data.oledb: Re: ODBC/OLE DB Connection Pool

And check out the exception noted above regarding the use of MTS. In ASP, if you use Server.CreateObject, you are forcing the use of MTS. Using vbscript's CreateObject will not cause MTS to be used and you may lose the

And this one (I've added numbers):

\*\*\*\*\*

Unintended Disabling of Resource Pooling

Resource pooling can be inadvertently disabled f