

## Re: ADO – permanent memory leak?!

**Source:** <http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.database.ado/2004-11/0034.html>

---

**From:** Rich S. (*Arby\_001\_at\_Yahoo.com*)

**Date:** 11/04/04

Date: 4 Nov 2004 09:25:25 -0700

I have a memory leak in an application that makes repeated calls to ADOQuery to update a statistics database. I was able to track the leak to this section of the code. I have ADO 2.7 SP1 installed on my Windows 2K PC. I am still running the Delphi 7 Enterprise. No update, the comments scarred me away for now.

I am using MemProof 9.5.0 to isolate the problem. And found similar messages to Jeff Dyer's concerning SysAllocStringLen. These are the only "memory growth" items.

Almost anytime I pass a string (widestring, AnsiString, variant, short string, or constant) to ADO I see these unreclaimed entries. This are normally items such as field names in Fileds(name) or paramByName(name), or data using .Value = somestring.

Searching the Internet has found others with similar problems dealing with ADO parameters that are Variants. I've also seen similar items in QualityCentral. Such as #8837 or 7070.

I would think that removing the file HookCom.Hlx only prevents MemProof from monitoring those calls. I don't think that it is reporting an erroneous error. I have not found any other references to this being a MemProof bug. If so I would appreciate a reference so I can better understand what is going on.

Is there another Memory profiler for Delphi that I could use to verify where the problem might be?

The problem seems to be inside of ADO, so I don't know why it isn't more of an issue. That makes me think it is something I'm doing.

For most applications the leak is small enough to not cause any issue. But for an application that is expected to run 24x7 and does a high frequency of table updates, it becomes a great

issue quickly.

Rich S.

"Michal R" <michal@topindex.cz> wrote:

>Thanks for your advice.

>It is as you have said. Deletion of HookCOM.hlx helped me.

>

>Michal R.

>

>"Jeff Dyer" <jeff.nospam@trustguard.co.uk> píše v diskusním příspěvku

>news:41792b4c@newsgroups.borland.com...

>> Michal – are you using MemProof 9.5.0? I just reported the same problem to

>> Atanas and it appears to be a fault in MemProof. I was advised to delete

>> the HookCOM.hlx file.

>>

>> Jeff

>>

>> "Michal R" <michal@topindex.cz> wrote in message

>> news:4178ec03@newsgroups.borland.com...

>>> Can somebody help me with ADO memory leaks?!

>>>

>>> Problem is in a few memory object which stays in memory – detected by

>>> MemProof:

>>>

>>> 1677 SysString 00168894 15

>>> SysAllocStringLen(ISAUTOINCREMENT,15)

>>> 1709 SysString 00168894 15

>>> SysAllocStringLen(ISAUTOINCREMENT,15)

>>> 1716 SysString 00168894 15

>>> SysAllocStringLen(ISAUTOINCREMENT,15)

>>> 1723 SysString 00168894 15

>>> SysAllocStringLen(ISAUTOINCREMENT,15)

>>>

>>> Source module is oleaut32.dll.

>>>

>>> When followed simple code is called in a loop, used memory by application

>>> is still growing till...:(

>>>

>>> A'm using D7 Prof with SP 1, Win XP and data provider is

>>> Microsoft.Jet.OLEDB.4.0 (Access DB).

>>> I also have tried with ODBC provider and MySQL server and problem is the

>>> same. The only one difference is in

>>> memory object which stays in memory (for access 22, for odbc/mysql 13).

>>>

>>> Thanks for help, Michal R.

>>>

>>> A have used this simple code:

>>>

>>> -----  
>>> procedure TForm1.specproc;

>>> var q:TADOQuery;

borland.public.delphi.database.ado: Re: ADO – permanent memory leak?!

```
>>> begin
>>> q:=TADOQuery.Create(nil);
>>> try
>>> q.SQL.Clear;
>>> q.Connection:=ADOConnection1;
>>> q.SQL.add('select * from mon');
>>> q.Active:=true;
>>>
>>> {...}
>>>
>>> q.Active:=false;
>>> finally
>>> q.Free;
>>> end;
>>> end;
>>>
>>>
>>
>
>
```