

## Re: Is file is being used

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-12/msg02655.html>

---

- *From:* [Eran.Yasso@xxxxxxxx](mailto:Eran.Yasso@xxxxxxxx)
  - *Date:* 13 Dec 2006 14:17:00 -0800
- 

Peter Duniho wrote:

<Eran.Yasso@xxxxxxxx> wrote in message  
[news:1166041591.215975.197100@xx](mailto:news:1166041591.215975.197100@xx)

Peter, nice to see pepole with the ammount of experience like you who reading posts of noobs like me.

We were all "noobs" at one time. For that matter, even the most experienced of us are \*still\* "noobs" in any variety of topics.

On to the subject...  
Why wouldn't access let more then one user access the file in the same time?

You'd have to ask the Access folks to get the exact details. But the general issue is that you can't safely have more than one arbitrary process writing to a file at the same time. There's no way to resolve how to coordinate the processes, and so the resulting file is non-deterministic.

For example, imagine two processes writing the strings "ABCDE" and "abcde" respectively, one byte at a time, to a file they both have open for writing at the same time. The file could wind up looking like any of the following:

- \* "ABCDEabcde"
- \* "AaBbCcDdEe"
- \* "ABaCDbEcde"
- \* "ABCabcdDEe"

or any other possible interleaving of the two strings. There's no way to allow both processes simultaneous access to the file and still have a well-defined consequence to them both writing to the file simultaneously.

The easiest, most straight-forward solution is to ensure that only one

## Re: Is file is being used

process can write to the file at a time. Once a process has closed a file, there's nothing it can do to ensure that no other process will change the file. But at least it knows while it has the file open, it can write to the file and leave the file in a well-defined state.

mdb file is not an ordinary file. think of it as multi file. it has tables. one can edit table while other edit other table. it shouldn't affect each other. Unless they edit the same table.

The file system doesn't know anything about the content of the .mdb file. It has no way to ensure that one process only makes changes to the file that affect one table, while another process only makes changes to the same file that affect a different table.

That said, most database do support multiple users, and they do so by providing an actual database server through which the database can be accessed. This adds a robust layer on top of the file itself that \*does\* define how different processes can modify the file at the same time, and which \*does\* understand the higher-level structure of the file (independent tables, for example).

Note that even in the case of a database server supporting multiple processes, it still enforces one-at-a-time access to any single part of the database, and there are still possible conflicts between different processes. Having the database server doesn't make the problems go away...it just abstracts them into a more usable scenario, where the manner in which processes cooperate is better-defined.

Pete

Pete, you are right if mdb file was an ordinary file. But, MDB file has tables. You can look at this tables as many different files attached to a big file. if both process don't write to the same table, then they are not "sharing the same memory"(file).