

# Why should I care about compacting an MS Access .mdb file?

Asked 16 years, 3 months ago   Modified 16 years, 3 months ago

Viewed 5k times



4



We distribute an application that uses an MS Access .mdb file. Somebody has noticed that after opening the file in MS Access the file size shrinks a lot. That suggests that the file is a good candidate for compacting, but we don't supply the means for our users to do that.

So, my question is, does it matter? Do we care? What bad things can happen if our users never compact the database?

ms-access

Share

Improve this question

Follow

asked Sep 16, 2008 at 16:53



Corey Trager

23.1k ● 20 ● 87 ● 121

9 Answers

Sorted by:

Highest score (default)



In addition to making your database smaller, it'll recompute the indexes on your tables and defragment

12



your tables which can make access faster. It'll also find any inconsistencies that should never happen in your database, but might, due to bugs or crashes in Access.



It's not totally without risk though -- a [bug in Access 2007](#) would occasionally delete your database during the process.



So it's generally a good thing to do, but pair it with a good backup routine. With the backup in place, you can also recover from any 'unrecoverable' compact and repair problems with a minimum of data loss.

Share Improve this answer

answered Sep 16, 2008 at 16:58

Follow



Nate

4,749 ● 2 ● 37 ● 45



5



Make sure you compact and repair the database regularly, especially if the database application experiences frequent record updates, deletions and insertions. Not only will this keep the size of the database file down to the minimum - which will help speed up database operations and network communications - it performs database housekeeping, too, which is of even greater benefit to the stability of your data. But before you compact the database, make sure that you make a backup of the file, just in case something goes wrong with the compaction.

Jet compacts a database to reorganize the content within the file so that each 4 KB "page" (2KB for Access 95/97)

of space allotted for data, tables, or indexes is located in a contiguous area. Jet recovers the space from records marked as deleted and rewrites the records in each table in primary key order, like a clustered index. This will make your db's read/write ops faster.

Jet also updates the table statistics during compaction. This includes identifying the number of records in each table, which will allow Jet to use the most optimal method to scan for records, either by using the indexes or by using a full table scan when there are few records. After compaction, run each stored query so that Jet re-optimizes it using these updated table statistics, which can improve query performance.

Access 2000, 2002, 2003 and 2007 combine the compaction with a repair operation if it's needed. The repair process:

- 1 - Cleans up incomplete transactions
- 2 - Compares data in system tables with data in actual tables, queries and indexes and repairs the mistakes
- 3 - Repairs very simple data structure mistakes, such as lost pointers to multi-page records (which isn't always successful and is why "repair" doesn't always work to save a corrupted Access database)
- 4 - Replaces missing information about a VBA project's structure

5 - Replaces missing information needed to open a form, report and module

6 - Repairs simple object structure mistakes in forms, reports, and modules

The bad things that can happen if the users never compact/repair the db is that it will become slow due to bloat, and it may become unstable - meaning corrupted.

Share Improve this answer

answered Sep 16, 2008 at 17:54

Follow



Chris OC

541 ● 2 ● 4

---

Re: queries and compaction, only the front end should have any queries in it, but front ends don't really need to be compacted. However, if the back end is compacted and table statistics change, compacting the front end will flag all saved queries as in need of recompiling the next time they run.

– [David-W-Fenton](#) Sep 17, 2008 at 3:18

---

Excellent points, David. I think in Corey's case, he doesn't have an Access front end so our posts are for general knowledge that applies to most Access developers.

– [Chris OC](#) Sep 17, 2008 at 10:45

---



4



Compacting an Access database (also known as a MS JET database) is a bit like defragmenting a hard drive.

Access (or, more accurately, the MS JET database engine) isn't very good with re-using space - so when a record is updated, inserted, or deleted, the space is not



always reclaimed - instead, new space is added to the end of the database file and used instead.

A general rule of thumb is that if your [Access] database will be written to (updated, changed, or added to), you should allow for compacting - otherwise it **will** grow in size (much more than just the data you've added, too).

So, to answer your question(s):

- Yes, it does matter (unless your database is read-only).
- You should care (unless you *don't* care about your user's disk space).
- If you don't compact an Access database, over time it will grow much, much, much larger than the data inside it would suggest, slowing down performance and increasing the possibilities of errors and corruption. (As a file-based database, Access database files are notorious for corruption, especially when accessed over a network.)

This article on [How to Compact Microsoft Access Database Through ADO](#) will give you a good starting point if you decide to add this functionality to your app.

Share Improve this answer

answered Sep 16, 2008 at 17:24

Follow



Keithius

1,124 ● 2 ● 12 ● 20

---

Compacting with ADO is only the best choice when not using Access as your programming platform. One thing this fine post omits is the distinction between front end and back end - back ends (data tables) need to regular compaction, while front ends do not (because they contain no data).

– [David-W-Fenton](#) Sep 17, 2008 at 3:15

---



1

I would offer the users a method for compacting the database. I've seen databases grow to 600+ megabytes when compacting will reduce to 60-80.



Share Improve this answer

answered Sep 16, 2008 at 16:58

Follow



[David Smart](#)

89 ● 1 ● 5



1

To echo Nate: In older versions, I've had it corrupt databases - so a good backup regime is essential. I wouldn't code anything into your app to do that automatically. However, if a customer finds that their database is running really slow, your tech support people could talk them through it if need be (with appropriate backups of course).



If their database is getting to be so large that the compaction starts to be come a necessity though, maybe it's time to move to MS-SQL.

Share Improve this answer

answered Sep 16, 2008 at 17:40

Follow



BIBD

15.4k ● 26 ● 87 ● 141



0

I've found that Access database files almost always get corrupted over time. Compacting and repairing them helps hold that off for a while.



Share Improve this answer

answered Sep 16, 2008 at 16:57

Follow



Danimal

7,710 ● 8 ● 48 ● 57



What do you do when you get to the point where it gets corrupted before it's compacted? I mean, you could restore from backup, but if you only did daily backups, and it would corrupt before the end of the day, how would you deal with such a system. As you said, compacting only holds it off so long. – [Kibbee](#) Sep 16, 2008 at 17:28



0

Well it really matters! mdb files keep increasing in size each time you manipulate its data, until it reaches unbearable size. But you don't have to supply a compacting method through your interface. You can add the following code in your mdb file to have it compacted each time the file is closed:



```
Application.SetOption ("Auto Compact"), 1
```

Share Improve this answer

answered Sep 16, 2008 at 17:26

Follow



Philippe Grondier

11.1k ● 3 ● 36 ● 73

- 
- 2 This is a terrible idea, as a compact on a database already in a suspect state can lose data that would otherwise be recoverable before the compact. Compact On Close should under all circumstances be turned OFF. – [David-W-Fenton](#)  
Sep 22, 2008 at 0:49
- 



0



I would also *highly* recommend looking in to VistaDB (<http://www.vistadb.net/>) or SQL

Compact(<http://www.microsoft.com/sql/editions/compact/>) for your application. These might not be the right fit for your app... but are def worth a look.



Share Improve this answer

answered Sep 16, 2008 at 17:36



Follow



calebjenkins

621 ● 5 ● 6

---

Neither of those things exist anymore. MSSQLExpress is the current recommendation for the backend of an Access DB.

[microsoft.com/en-us/sql-server/sql-server-downloads](http://microsoft.com/en-us/sql-server/sql-server-downloads)

– [HackSlash](#) Apr 13, 2020 at 19:56

---



-1



If you don't offer your users a way to decompress and the raw size isn't an issue to begin with, then don't bother.

Share Improve this answer

answered Sep 16, 2008 at 16:56

Follow



EndangeredMassa

17.5k ● 8 ● 56 ● 80



