

# 10 ways to prevent Access database corruption

**Date:** April 1st, 2008

**Author:** Susan Harkins

**Category:** 10 things, Tips

**Tags:** Microsoft Access, Workstation, Network, Data, Susan Sales Harkins, Storage, Databases, Hardware, Enterprise Software, Software

**By Susan Sales Harkins and Gustav Brock**

Nothing frustrates the full spectrum of Access users — from casual user to developer — quite like a corrupted database. With a little know-how, you might get lucky enough to repair the database or at least recover the data. Sometimes, a third-party product that specializes in recovering data from a corrupted database can help. But you could end up rebuilding and re-entering data. (Or rather, your replacement will rebuild and re-enter it.)

A more proactive approach to the situation is to avoid corruption in the first place. Here are some strategies for preventing your databases from becoming corrupted.

*Note: This information is also available as a PDF download.*

## #1: Split your database

If more than one person will access the database, split it into two pieces: a backend that stores the data in tables and a front end that contains everything else (forms and reports). Access has a wizard that holds your hand through the process so there's no excuse not to split a shared database. Name the two ends appropriately. Remember, the backend contains your "gold" — your data. With the data and interface objects in separate databases, you can easily replace the front end from a backup, if necessary. In addition, this setup makes enhancements easier to incorporate into your system.

There is some discussion about whether the front end should be stored on a network server (and shared) or on a local drive. This decision really has no impact on corruption, so the choice is yours. If possible, consider write-protecting a shared front end. If the worst happens and something destroys the front end, simply obtain a fresh copy of it to replace the broken one and reboot the local system.

## #2: Store temporary tables in a backend

If a database creates, populates, and then deletes temporary tables, keep those tables in a separate backend database to prevent bloat. Name the additional backend accordingly, making its purpose obvious. This additional backend can be shared or local.

## #3: Don't use memo fields

Avoid using memo fields if possible. They often, indirectly, cause corruption. Even though the database in these cases often can be repaired, some content in the memo fields might be lost. If you need memo fields, keep them in separate tables and create a one-to-one relationship to the parent table(s). Even better, move memo tables to a separate backend database file and name the file accordingly, to indicate its purpose.

#### **#4: Don't store picture files**

Usually, you shouldn't store picture files in a database. If you must, treat them the same way you would a memo field (see #3). Access has no problem attaching tables from multiple backend databases.

#### **#5: Create temporary tables to speed up queries**

If you run complex or nested queries (where one query pulls data from others that hit still others), Access may write a lot of temporary data that you never see. Most often, this happens when a query that works with a small amount of data performs slowly, putting stress on the JET engine. If Access chokes during this process, you can end up with a corrupt backend file.

To prevent this problem, write some of the temporary data to temporary tables. There's no universal method to recommend. Analyze the specifics and run some tests to find the best solution. However, sometimes the use of just one temporary table can minimize the chance of corruption *and* speed up the queries by a factor of 10 or more.

#### **#6: Be careful with wireless networks (WiFi)**

A connection may work fine, but multiple users or powerful neighborhood networks (or other noise sources) can abruptly cut off the connection. That can corrupt the database file if you are writing to it at the time. This type of interference isn't a problem if users are mostly reading from the database.

#### **#7: Be careful with WAN connections**

A WAN connection that covers any connection from a local system to a server via the Internet can cause trouble. Reading the database may be slow but acceptable. However, writing to the database is error prone and can cause corruption. When bottlenecked traffic interrupts data transfer, Access times out, believing the connection has been lost. This behavior usually leaves the backend database in a corrupted state.

#### **#8: Don't put Mac and Windows users on the same network**

If Mac and Windows users share the same network and experience problems, establish a separate network for database users. Macs are extremely noisy, and typical Mac applications generate vast amounts of network traffic when moving large graphics files and printing. In a shared environment, use only high-quality network components.

It's best to keep the database traffic separate from the graphics traffic, as much as possible. You can accomplish this by allowing administrative workstations to connect directly to the server with the shared database via a local switch.

## #9: Troubleshoot network hardware

If corruption just happens from time to time, you may have to deal with a network hardware issue. First, try to narrow down the workstation and swap the error-prone station with another. If the issue follows the workstation, you know that's the source. It might be easiest to get rid of the workstation.

If the problem isn't specific to the workstation, the error is most likely to be found in the connection from the workplace to the network switch — including the actual switch port. If the cable's in good shape, swap the ports between the workstations. If the error source now swaps too, replace the switch; if not replace the cable. If cables aren't permanently installed or don't cross from floor to floor, you can try replacing them before swapping workstations.

## #10: Check the server's configuration

Sometimes, the configuration for the server where you've stored the shared backend database file is the culprit. You'll need a specialist to track down and resolve this type of problem. There are several possibilities, from server parameters to a malfunctioning disc controller to a misconfigured RAID array.

You can temporarily move the backend file to a different location, like one of the popular NAS devices or a shared folder on a workstation. If the problem with corruption goes away, call the specialist. If not, the corruption's source is somewhere else (#1 through #9).

---

Susan Sales Harkins is an independent consultant and the author of several articles and books on database technologies. Her most recent book is *Mastering Microsoft SQL Server 2005 Express*, with Mike Gunderloy, published by Sybex. Other collaborations with Gunderloy are *Automating Microsoft Access 2003 with VBA*, *Upgrader's Guide to Microsoft Office System 2003*, *ICDL Exam Cram 2*, and *Absolute Beginner's Guide to Microsoft Access 2003*, all published by Que. Currently, Susan volunteers as the publications director for Database Advisors. You can reach her at [ssharkins@gmail.com](mailto:ssharkins@gmail.com).

Gustav Brock is a consultant and developer at Cactus Data ApS, an independent software house specializing in accounting and time billing systems and custom database applications. He has been working with Microsoft Access since version 1.0 and is a frequent contributor to the forum of developers hosted by Database Advisors.



**Susan Sales Harkins** is an IT consultant, specializing in desktop solutions. Previously, she was editor in chief for *The Cobb Group*, the world's largest publisher of technical journals.

## People who read this, also read...

10+ reasons why IT pros hate Microsoft Access (but really shouldn't)  
Reduce mouse clicks with these two Access options  
10+ reasons to use Access (and a few reasons not to)  
10+ things you should do before building a custom Access database