

Preventing corruption of MS Access databases

The easiest way to corrupt an Access database is to interrupt a write operation. The **inelegant disconnection** accounts for almost all corruptions. If you experience corruption on a database that is not being modified (i.e. only data is changing, not forms, reports, ...), check these areas:

| Area | Cause of Corruption | Preventative Action |
|---------------|---|---|
| User | Switching computer off without closing Access. (Users rarely own up to this.) | Train users. Log users in/out of the database to see which users/computers are crashing out. |
| Power | Supply issues: spikes, dips, brown outs, outages, or tripping over the power lead. | Consider a power conditioner or UPS for each workstation. |
| Hardware | Overheating, inadequate power supply, intermittent components, device conflicts. | Troubleshoot and replace bad component/computer. |
| Software | Any application crashing and bringing down others. | Ensure the latest service packs are installed. Remove unnecessary software. |
| Network | Intermittent cable connectors, hubs/switches, network cards, inherently unstable networks such as WiFi. | Troubleshoot and replace network hardware. |
| Configuration | Sharing a single mdb file amongst users. | Split the database into front end/back end, so each user has their own copy of the front end. |
| Full disk | Insufficient hard disk space for the temporary folder and/or virtual memory. | Check the temp environment variable is valid, and the hard disk has hundreds of megabytes free. |
| Service Packs | Workstations on different service packs (particularly JET) can cause inconsistent behavior. | Download the Office service pack and the JET 4 service pack from http://support.microsoft.com , and apply to all workstations. |

Some basic steps can lessen the chance of serious loss: backups, periodic Compact/Repair, and simply closing the database when not in use.

Preventing corruption during development

Your database will corrupt during development when objects and code are constantly being designed and tested. These tips will reduce but not eliminate corruption:

1. Install the latest proven **service packs** for your version of Access and JET.
2. Turn off **Name AutoCorrect** before creating any objects in your database. This thing is [buggy](#), incomplete, performance degrading, and contributes to corruption. Turn it off under File | Options | Current Database.

Preventing corruption of MS Access databases

3. In your **subforms**, include a **text box for each field** in its *LinkChildFields* property. This simple step avoids frequent of Access 2002 and 2003 ("... shut down by Windows..."), where the form is based on a query (not directly on a table.) The text boxes can be hidden if you do not wish to see the foreign key fields.
4. Never allow **multiple developers** to modify the objects in a **single file at the same time**. Give each one a local copy, and import their objects back to the main copy later. This tip alone will save you much grief.
5. Do **not edit** the text of a module **while it is running** (i.e. in break mode). If a project is compiled, and you edit the text while it is running, the Compile item on the Debug menu may remain greyed out. This may indicate that Access has become confused between the different copies it keeps of the code: the compiled code, the text-version of the code, the text-version of the temporary copy that it creates when you edit the form's module, and the version that is actually running. Refusing to edit code in break mode significantly reduces corruption in my experience.
6. Watch out for the first **hint of corruption**, and take action. A project is beginning to corrupt if the Compile doesn't hold. If a compiled project is uncompiled after a restart, it is time to backup, compact, **decompile**, compact again, compile, and make another backup copy.
7. If Access crashes, do not allow it to compact and reopen. Delete the ldb file. Create a copy of the crashed database. Then reopen, and compact/repair. If it did corrupt, it may be possible to rescue some objects from the corrupted database provided it has not been compacted. The undocumented commands *SaveAsText* and *LoadFromText* are useful in this context.
8. Create a **backup** copy and **compact** regularly during development. Our pattern is twice a day, so we never lose more than half a days work.
9. In your code, **close everything you open** and **dereference your objects**. If you do not open something (such as the default Workspace, or the RecordsetClone of a form), do not close it, as that can cause problems. If you do open something (such as OpenRecordset), close it. Then set all object variables (Form, Recordset, TextBox, ...) to Nothing.
10. Be especially careful with any code that does not have the normal **safety net** VBA provides, e.g. Eval(), API calls, late binding.
11. Split the database, and give users an executable version that cannot decompile:
 - In Access 2010: File | Save & Publish | Make ACCDE.
 - In Access 2007: Make ACCDE on the Database Tools ribbon.
 - Earlier versions: Tools | Database Utilities | Make MDE File
12. Unless you really need it, turn off Record-Level Locking (as opposed to Page-Level Locking.) Page-Level locking is more efficient to execute, and works more reliably in some circumstances. In Access 2000 - 2003, uncheck the box under Tools | Options | Advanced | Open Databases Using Record-Level Locking. In Access 2007 and later, go to File (Office Button) | Options | Advanced.