



## Road to recovery

**Damaged database? Lost information? Mark Whitehorn forms a one-man anti-corruption squad to root out repair options and ways to recover lost information.**

In my experience, well written, single-user Access databases are very reliable, and corrupt databases are rare. Corruption becomes more common — but still very infrequent — when multiple users access the same database file. Of course, it is perfectly possible to write an Access application so badly that it actively mangles the database. But, presumably, since we are all too smart to actually ever do that, it is never a problem in real life. Well, not entirely. On rare occasions Access databases do become corrupt and I've received several letters asking about ways to recover information from such databases.

### Single file

An Access database stores everything in a single .MDB file — that's tables, data, queries, forms, reports, code, macros... the lot. The internal structure of this file is horribly complex and although the specifications may have been published, they are certainly not well understood. To add to the complexity, the structure changes with every upgrade (BTW — the most recent version, Office 2000, for the first time in history can actually write files in the previous version). It is fairly common, however, particularly with multi-user databases, to store the data in one .MDB file and store the rest in another. I recommend this strategy, but for the purposes of this article I shall presume a single .MDB file: in

practical terms there's little difference between dealing with either arrangement. Damage to a database can affect either the data or the components that comprise the database structure, or both. Occasi

***I let it run for a while, but my nerve finally ran out and I killed the print job at 7000-plus pages***

about saving the old backups (essentially making them incremental) using a simple naming scheme like 10DEC1998.MDB. This will provide a series of snapshots of the database at particular points in time. So, imagine that you have a database

onally, components can become broken in relatively subtle ways.

One I experienced recently was a form that appeared perfectly normal when it was used to look at data, but whenever it was used to print records, it printed those requested and then continued to print an apparently infinite set of blank pages. I let it run for a while to see where it would stop, but my nerve finally ran out and I killed the print job at 7,000-plus pages. Before Greenpeace put me on a hit list, these were virtual pages in the print queue. It goes without saying

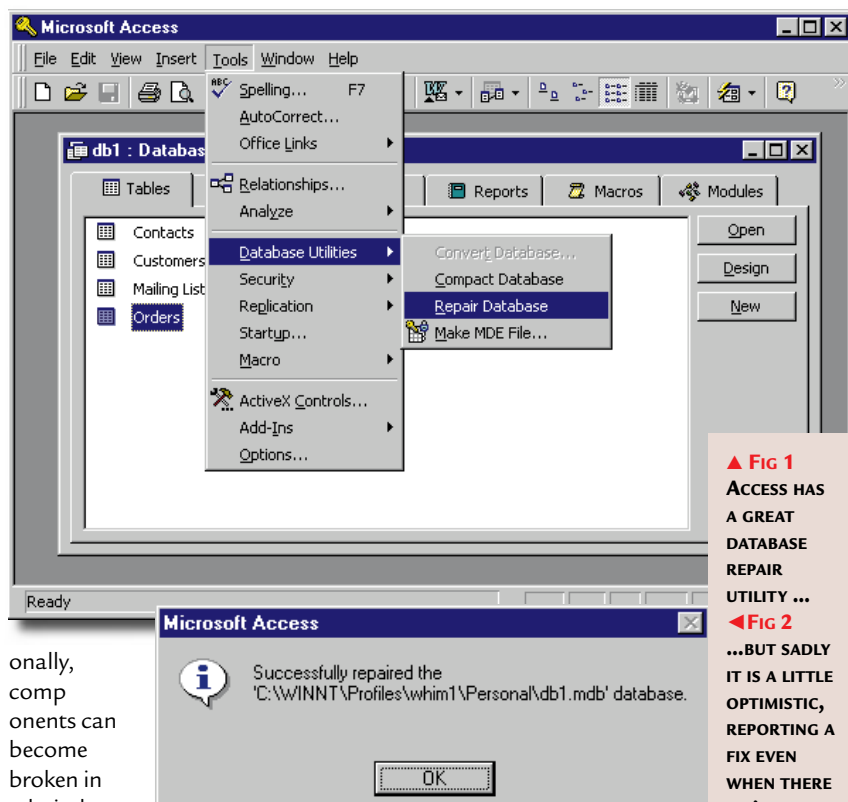
that you will be making regular backups to disk or tape, of any and all MDB files that you use. But think

which opens and all of the components work, except one, say a form that has gone rogue. What can you do?

➤ **Immediately** make two or three copies of the damaged database and work on one of these. That way, things won't get any worse. If the form is simple, just delete it and recreate it.

➤ **But what if** the form is complex and would be time-consuming to rebuild? Try running Repair Database from the Tools, Database Utilities menu and see if that helps [Fig 1]. The repair utility will tell you if it can't fix a problem, but blithely reports that it has successfully repaired the database [Fig 2] regardless of whether it found anything to repair or not!

➤ **What if** that didn't work? Assuming that you have incremental backups, work your way back through the files until you reach a working version. There's no absolute path to follow now, but working logically and methodically you can swap-in the working component



▲ **Fig 1**  
ACCESS HAS  
A GREAT  
DATABASE  
REPAIR  
UTILITY ...

◀ **Fig 2**  
...BUT SADLY  
IT IS A LITTLE  
OPTIMISTIC,  
REPORTING A  
FIX EVEN  
WHEN THERE  
ISN'T  
ANYTHING  
WRONG. THIS  
MEANS YOU  
CAN'T TELL  
WHETHER IT  
HAS FOUND A  
PROBLEM,  
LET ALONE  
FIXED IT

from this .MDB file to replace the rogue one. Incidentally, queries are among the easiest to swap as you can open one in design mode, go to the SQL code and cut and paste that. Since this is pure text, it usually transfers well. As another aside, before anyone tells me that you can't

## Most of this comes down to using common sense and trying not to panic

open two .MDB files at once in Access — I know. This makes cutting and pasting between .MDBs a pain. However, there is nothing to stop you running two instances of Access on the same machine and flitting between them with Alt-Tab.

➤ **Think about** what is vital in the .MDB file. This is usually the data, so try exporting it as text files. That is relatively straightforward, and if it can be achieved, you can relax a bit. Then create a new .MDB file and try to import the tables into that from the damaged .MDB file. If you can do that, then you have rescued not only the data but also the data types etc.

➤ **Next, start moving** components, one at a time, from the damaged to the new .MDB. Clearly it makes sense to try to identify the damaged component(s) and move everything but these. It also makes sense to check that everything is working at each stage of the replacement to ensure you haven't reproduced the problem. If the database has hundreds of forms, reports and so on, copy them in batches and continue to test to see if you've brought in a duff component.

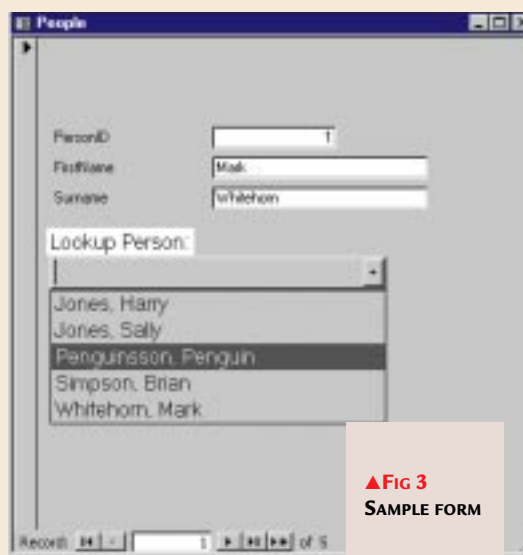
## Don't panic!

To sum up, most of this recovery process comes down to using common sense and trying not to panic — and keep taking the incremental backups.

## PCW CONTACTS

Mark Whitehorn welcomes readers' suggestions and feedback for the Databases column. He can be contacted via the PCW editorial office (address, p10) or email [database@pcw.co.uk](mailto:database@pcw.co.uk)

## THE CRAFTY COMBO BOX



▲ Fig 3  
SAMPLE FORM

The following combo-box tip comes from Chris Veness [chrisv@movable-type.co.uk](mailto:chrisv@movable-type.co.uk).

"The standard 'navigation buttons' normally displayed at the bottom of forms generally correspond to working through an unsorted list one by one. I often find a better way of allowing users to find records is the combo box.

Take a form showing information about people and assume the form is based on a table

which includes fields for PersonID, FirstName and Surname. What I generally do is create an outsize combo box named something like LookupPerson (I also make it the same colour as the form, and raised, so that its appearance distinguishes it from fields used for entering information). The Control Source is left blank so that it is 'unbound' and the Row Source is set to something like

```
SELECT PersonID, Surname & ", " & FirstName AS Name
FROM Person
ORDER BY Surname & ", " & FirstName;
```

"The Column Count should be 2, the Bound Column 1, and Column Widths set to 0cm to hide the first column (the width of the second column doesn't need to be specified). It's best to set Limit To List to Yes and AutoExpand to Yes, and probably to leave the List Width at Auto. This creates a pick list that will show all the people in the database, in alphabetical order.

"Now we have to make it look up the details for the person selected from the list. This is done by associating an event procedure with the After Update event of the combo box. This will be along the lines of:

```
Private Sub LookupPerson_AfterUpdate()
Me.RecordsetClone.FindFirst"PersonID = " &
Me!LookupPerson
Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub
```

"FindFirst won't operate directly on a form, so we have to use the RecordsetClone property of the form. Doing a FindFirst on the RecordsetClone won't affect the form until we synchronise it by setting the form's bookmark to the RecordsetClone's bookmark. Not only is this pick list easier to use, the auto fill means that the user can type the first few letters of the name and the pick list will automatically move to the matching entry."

*Nice one, Chris! I have built a sample form [Fig 3] based on Chris's idea and it's on our cover-mounted PCW CD-ROM this month. Admittedly, I know it is in Access 97, not 2.0, but unfortunately the portable on which I am writing this doesn't have Access 2.0. However, all I did was follow Chris's instructions, so Access 2.0 users should be able to whip up the same sort of thing in about five minutes. Bear in mind that while this combo box works beautifully for small sets of records, it will get slower as record numbers increase.*