GOLDSTAR SOFTWARE
www.GoldstarSoftware.com

Tech Tip #18: File Corruption Recovery

Overview: BUTIL Recovery Process and LowLevel Tools

In our last Tech Tip, we saw what corruption looks like to the application, to the engine, and to the data file. In this Tech Tip, let's see how to fix it.

A database file is similar to a three-ring binder – it contains data pages, to be sure, but it also contains index pages and other important information so that the database engine can locate the data within it. If someone puts "bad" data into the system, such as an invalid date, the engine may actually handle this perfectly – the old example of garbage-in, garbage-out. This is what we call "*data* level corruption", and while it may cause the application to fail, it is not "*database* level corruption". An example of database corruption would be if a truck ran over your binder, and you could no longer access all of the data that is supposed to be contained therein.

If your binder is run over, the process to fix it is simple. First, you survey the damage, so that you know how much work to expect. You see one crushed binder. Then, you collect as much of the raw data as you can. If you're lucky, you can just pick up the crushed binder and take out the pages. If not, you may have to take more drastic action -- running through traffic to collect the pages blowing in the wind. Once you have as much data as you can recover, you go to the store and buy a new binder with many of the same attributes as the original – three rings, standard size, etc. Finally, you put the data back into the new binder, and you're all done.

With a database file, the process is the same: we recover the data, get a new empty binder, and put the data into the binder. The tool that we use for this is **BUTIL**. Let's look at the three steps as we recover a sample file called CORRUPT.MKD:

1. **Survey the damage**: We use the **STAT** function to get file statistics from the database file:
   ```
   BUTIL –STAT CORRUPT.MKD
   ```
   Record the number of records in the file, so that you will know if the process is successful.
2. **Recover the data**: The **RECOVER** function aims to export all of the data from a damaged file and put it into a UNF file. (A UNF file is a flat file format designed to hold only raw record blobs, and no key definitions or anything else.) The process is not always successful, but often times it works just fine. We recommend using the /I and /J switches to recover records at the start and end of the file. To export the data, the command for our sample file is:
   ```
   BUTIL –RECOVER CORRUPT.MKD CORRUPT.UNF /I /J
   ```
   When the process finishes, it will report a total number of records recovered. Now is the time to compare this number with the number reported in Step 1. You can then decide if it is worth continuing this process, or just restore the files from backup instead.
3. **Make a new binder**: All of the file statistic information is stored in the File Control Record, which is at the beginning of the file. As long as the FCR is readable, we can use the **CLONE** function to create a new database file that looks just like the original. This command is easy and quick:
   ```
   BUTIL –CLONE NEWFILE.MKD CORRUPT.MKD
   ```
   Note that the new file will be created in the file version indicated by the *Create File Version* configuration setting. If you need it in a different format, you may need to use **Rebuild**.
4. **Put the data back**: Finally, the last step is to put the data (currently housed in the UNF file) back into the new database file. We do this with the **LOAD** function:
   ```
   BUTIL –LOAD CORRUPT.UNF NEWFILE.MKD
   ```
   This process loads the records back into the file. Here, the important thing to watch for is the total record count. Again, if it matches the number in Step 1, you have done a perfect recovery!

Once this process is complete, the last step is to throw away the old binder and put the new one in its place. This is done by deleting the damaged file and renaming NEWFILE.MKD to CORRUPT.MKD. If you need more information about this process, check out our white paper on Data File Maintenance:

http://www.goldstarsoftware.com/whitepapers-troubleshooting.asp

What do you do if the recovery process doesn't work and your backup is months or years old? Well, the first step would likely be to polish up your résumé. Beyond that, you may want to consider bringing in the experts to help recover your data. Goldstar Software has a **Data File Recovery Service** that may be able to recover more data than the standard **BUTIL** tool can get out. You can find out more information about this service from our web site here:

http://www.goldstarsoftware.com/recovery.asp

If you are a developer and your users are constantly getting into hot water with corrupted files, then you might find it worthwhile to learn how to do this in-depth file recovery on your own. Goldstar Software sells our **Pervasive Data File Recovery Toolkit** – a complete package that provides the same diagnostic and file recovery software that we use internally for data file recovery. In addition, the **PDFRT** package includes documentation on the underlying file structure, as well as information on how to use the various recovery tools to recover your data.

http://www.goldstarsoftware.com/pdfrt.asp

All in all, data file recovery is a highly technical process that is not for the novice or for the faint of heart. However, if you are developer with a solid understanding of hexadecimal numbers, data structures, database coding, and file systems, and if you are game for learning how to handle the recovery process on your own, these tools are the best in the industry!