

# FileMaker Database Corruption Example

Ebase SOLICIT\_.103 File

Larry Bednar

June 20, 2003

## Purpose

This document provides a description of an interesting example of FileMaker database corruption and the process used to identify and correct it.

## Overview

In June 2003, the Oregon Natural Desert Association (ONDA) attempted a transfer of data from their historic ebase V1.0.0 application into a fresh set of ebase V1.0.3 files. After transfer of data into V1.0.3 files using my “ebase data transfer tool” (EDTT), they discovered that the database crashed in certain situations involving the use of a drop-down value list on a field displayed within a layout of the SOLICIT\_.103 database included in ebase V1.0.3. No problem of this kind had been seen during use of ebase V1.0.0.

The symptoms were revealed only in the following circumstance:

1. Open Ebase V1.0.3
2. Open the SOLICIT\_.103 database
3. Open the “SC Maintenance – Edit” layout
4. Place the cursor within the field labeled “SOURCE CODE” at the top center of the layout  
(This is a display of the “SOLICIT\_.103::Display Source Code” field, using a pop-up list based on the value list “Source Codes”.)
5. Scroll downward in the pop-up list of values provided for the “source” field
6. When the cursor nears the end of the list, FileMaker crashes.

ONDA provided me a copy of their ebase V1.0.0 application. These files were still in the original FileMaker V4 format used with ebase V1.0.0.

The nature of the problem seemed to indicate possible database file corruption as the cause. But it wasn't clear whether the problem was related to the data import operations, conversion of files from FileMaker V4 to V5 format, corruption already present in the V1.0.3 files used to accept data from the V1.0.0 database, etc.

The problem was finally attributed to data-derived corruption. The problem never recurred after several records were deleted from the SOLICIT\_.103 file.

## Troubleshooting/Resolution Steps

Because the data transfer involved a conversion to from both FileMaker V4 to FileMaker V5 and from ebase V1.0.0 to ebase V.1.0.3, both FM V4-V5 conversion and the data transfer from ebase V1.0.0-V1.0.3 required investigation.

The following sequence of actions was applied to identify the problem cause and correct it:

1. Rename provided SOLICIT\_.100 file to SOLICIT\_DMG.100.

2. Open FM V4 version of Ebase V1.0.0, and also open SOLICIT\_DMG.100
3. Verify apparent correct operation of SOLICIT\_DMG.100  
This verified that there was no obvious problem manifest in the ebase V1.0.0/FM4 SOLICIT\_.100 file.
4. Convert SOLICIT\_DMG.100 to FM V5 format
5. Open FM V5 version of Ebase V1.0.0, also opening the FM V5 version of SOLICIT\_DMG.100
6. Verify apparent correct operation of FM V5 version of SOLICIT\_DMG.100  
This verified that there was no obvious problem manifest as a result of conversion of the SOLICIT\_.100 file from FM4 to FM5 format.
7. Export data from FM5 version of SOLICIT\_DMG.100 to new FM5 database named SOLICIT\_DMG\_EXPT.100
8. Open a copy of Ebase V1.0.3 created from the EDTT tool
9. Verify correct operation of SOLICIT\_.103 database within "fresh" copy of Ebase V1.0.3  
This verifies that there was no obvious problem with the fresh copy of Ebase V1.0.03 into which data was imported during the "data transfer" operation.
10. Manually import data from FM5 version of SOLICIT\_DMG.100 to "fresh" copy of SOLICIT\_.103
11. Test SOLICIT\_.103 for corruption - db crash results  
This verifies that the crash was caused by the data imported into the fresh SOLICIT\_.103 file, which operated correctly prior to the data import.
12. Recreate a "fresh" copy of all Ebase V1.0.3 databases from WinZip archive  
This step was taken to ensure that the copy of SOLICIT\_.103 used in further testing was not corrupted by the crash at step 11.
13. Verify correct operation of SOLICIT\_.103 database within "fresh" copy of Ebase V1.0.3
14. Manually import data from FM5 version of SOLICIT\_DMG\_EXPT.100 to "fresh" copy of SOLICIT\_.103
14. Create/run scripts in SOLICIT\_DMG\_EXPT.100 that open each field and each record using "all fields" layout - no problems revealed with data  
This is a method of testing for data corruption suggested by Eric Johnson on the ebase mailing lists in the summer of 2003.
15. Test SOLICIT\_.103 for corruption - db crash results (NOTE that crash results in all cases when records near the end of the scroll down list for "source code" at the top of the SOLICIT layout displayed by ebase at file opening)  
This demonstrates that the crash is results only in a specific *usage* of the data.
17. Recreate a "fresh" copy of all Ebase V1.0.3 databases from WinZip archive
18. Verify correct operation of SOLICIT\_.103 database within "fresh" copy of Ebase V1.0.3
19. Check source of drop-down value list causing problems

20. Manually import data from FM5 version of SOLICIT\_DMG\_EXPT.100 to "fresh" copy of SOLICIT\_.103
21. Create/run scripts in SOLICIT\_.103 which cause each field and each record to be opened using "all fields" layout - no problems revealed with data
22. Test SOLICIT\_.103 for corruption - watch closely for EXACTLY which part of value list is being visited when db crashes (NOTE that crash seems to result when a source code with text "renew..." is being displayed)  

This was taken as an indication of which specific data *records* were causing the crash.
23. Recreate a "fresh" copy of all Ebase V1.0.3 databases from WinZip archive
24. Verify correct operation of SOLICIT\_.103 database within "fresh" copy of Ebase V1.0.3
25. Perform a "find" in SOLICIT\_DMG\_EXPT.100 to exclude records with the text "renew" in field "Source Code" or "Source Code Description"
26. Manually import data from FM5 version of SOLICIT\_DMG\_EXPT.100 to "fresh" copy of SOLICIT\_.103, and verify that "renew" records have not been imported
27. Test SOLICIT\_.103 for corruption - no problems encountered

The resulting copy of the SOLICIT\_.103 file is the one I sent to ONDA. They never again saw this problem. This seems to clearly indicate that the problem was data-derived. ONDA manually reentered the records that I excluded from the import at steps 25/26.

## Discussion

It is interesting to note how specific the manifestation of this data-derived corruption was. Only in one specific use of that data was a crash caused. It was possible to look at that same records using other approaches (as in step 21 above) with no apparent problems manifest. This is a clear example of how database corruption present in a FileMaker database may not be noticed until much later – and even then only in very limited circumstances. This lends weight to the usual recommendation that any crash of a FileMaker database should lead to a “revert to recent backup” approach by the database administrator.