MF Tech Notes

November 17, 1993 - Carl Brown

Q: The VB Sample wont run.
A: Get VBRUN300.dll. It's available just about everywhere.

Q: Lock and Unlock don't work as documented.
A: If you are not on a network, you must run SHARE.exe to use the Lock/Unlock calls. Some networks also require 'share' to be loaded.

October 1, 1993 - Carl Brown

Q: My indexes don't seem to be in sorted order... Any idea as to why?

A: The most common problem for MF has been the mfCreateDB function. When something seems to be out of wack -- the database or indexes were probably created improperly.
Things to check for:
- Array size for INDEXes is large enough to hold all the values required

- FILENAME of the database should NOT have an extension

- The correct sort-key was used when creating the database/index (MFCOMP_CHAR, ...)

- Your record 'structure' matches how you created the database. (i.e. Indexes are FIRST in the structure and match the parameters you created the indexes with).

These are the most common problems. I'm sure there are many other reasons the indexes are not sorting. It may well be that the 'indexes' are broken (due to a power-failure, read-failure, or some other uncontrollable act). New to this release is the mfReIndex command. If you think the indexes are corrupt, you can reindex and see if the problem goes away. If it doesn't go away -- send the source code and we'll have a look at it.

Q: Why doesn't MF have an 'extension' on the file and why can't I add one to it?

A: At first look, this seems like a natural thing to want. However, there is one interesting problem with this:
- A future release (2,0?) will automatically split databases and indexes. We need those extra bytes to allow for naming the split files. Since a great deal of effort hasn't been expended on thinking of all the possible combinations and how we might lay this out -- we had always intended to use the next byte. Therefore, you databases would be named like:
MyDB.
MyDB.a   - The database -- part 2...
MyDB.1   - Index 1
MyDB.1a   - Index 1 - Part 2
MyDB.1b   - Index 1 - Part 3

That leaves us with a one character extension. However, if we ever support more than 10 indexes per DB, we would have the problem of:
MyDB.10a

Which doesn't leave any extra characters.

An alternative approach would be to use an extension ONLY on the database itself (and not the indexes...). Again, this hasn't been hammered out and very little planing has been placed in the split data and indexes. We also hope that a future release will have a:
MyDB,MF file.
This file would have a 'field' list of everything in that DB and allow stand-alone editor to open the database and browse the fields.

Q: mfInit vs. mfOpen... Should mfOpen be called once for each mfInit?

A: No. mfInit should only be called one time per application instance. (See next tech note for interesting uses of multiple calls to mfInit).

May 10, 1993 - Carl Brown

Q: MF doesn't support as many files as I would like it to. (e.g. only 14 DB's can be open at once). Can I register another 'task' with MF so I can get 14 more DB handles?

A: Yes. You can register another task. In the next release (or as soon as possible) we will make MORE file handles available per task. However, keep in mind that: Windows can't have TOO many files open before it runs out of file handles. Windows defaults to 20 FILES per application. You can increase this using the Win API SetHandleCount(iNumOfHandlesNeeded).

As an alternative, MF is VERY fast at opening/closing files. If you are using most of the DB's intermittently, we think you'll be surprised at how fast MF will open, read, and close a file...

Q: Is there any type of RAW data editor so I can 'browse' my databases?

A: It's in the works. Along with a 'DDL' (data description language) that will allow you to 'design' the database and work with 'field names'. The DDL will also allow 'browse' tables to be built and displayed through a custom control.

Q: Do disk-doublers affect MF?

A: MF has been tested on DBLSPACE and STACKER. No problems have surfaced. However, you will suffer a SEVERE performance decrease when using a disk doubler with MF. (Actually, all databases suffer it... not just MF). Generally, your apps will run about 30% as fast as w/o a disk doubler installed. (This only applies to writing data -- reads will be about the same speed.)

Q: I run my program 10-15 times and then -- it locks up. Why is this?

A: One of two reasons:
1- You are not calling mfDeInit before exiting your application and you are running out of file handles.

2- Same as first reason, only worse: You are in VB-RUN mode or a DE-BOOGER and are aborting without calling mfDeInit.

All things being equal, mf.DLL will automatically re-initialize itself IF it gets a FREELIBRARY call. This will occur when: a) a VB application ends or b) a C application calls FREE LIBRARY (however, if you link the mf.lib file, then you don't need to call FreeLibrary or LoadLibrary...).

In either case, if an application terminates abnormally, MF.DLL is left in memory and it doesn't realize the calling application is no longer active. It continues to give you a new task # until it runs out of tasks to give out.

There are two ways to get a DLL out of memory: 1- Restart Windows, 2- Call FreeLibrary (with the appropriate parameters) until it returns an error. A utility is available in the MS-SDK section on CompuServe called KillDLL. It's freeware and will dump a DLL from memory that you don't want there. (It was written by us).

April 26, 1993 - Carl Brown

Q: Can I rename the MF.DLL to something else and still use it? Our company wants a completely in-house look to it and we don't want other companies names on our product.

A: Yes. However, you may only do this after you have paid for the distribution rights. If you wish, we will even stamp the DLL with your company name. (The internal description of the DLL...)

Q: Does MF.dll HAVE to be in the windows\system directory?

A: No. It can be in the windows directory instead! <vbg> This question stems to a lot of problems in windows. You can place a .DLL anywhere AS LONG AS you tell windows WHERE it is when you load it. This is a problem (sometimes) in VB if you are in 'Interactive' mode. VB won't always find a DLL in the directory with the rest of your programs if the place that the code is running is not in the system path. Did that make sense?

OK, with that in mind, these are the places you can put MF.DLL (and, usually, any other DLL...)

- Anywhere in the DOS PATH
- In the directory that the EXE accessing it will be loading from
- ANYWHERE if you use the WinAPI LoadLibrary (with a full-path to the DLL)

MS has mandated that all shareable files should be placed in the Windows System directory. It's up to you if you want to follow this advice.

Q: I am developing an application in C and VB. The part in C is a DLL that is called from VB. Do I have to call mfInit for the DLL (in C) and the actual application in VB?

A: No. However, you should call mfInit from the VB part (or, the actual .EXE) and pass that task handle (returned from mfInit) to the DLL for use by the DLL to call MF. VB and C can both call MF without any problems. This allows you to place all the 'thinking' parts in C and the user interface and 'simple' stuff in VB. It is important to keep in mind that you should have a unique 'task' handle for every application (and/or instance) that will access MF. In other words, if you are creating a 'DLL' that will be used by others and that DLL will access MF directly, you should require the 'callers' .EXE's to pass you a task handle that can be turned into a MF TASK handle.

e.g.

Lets say you write a 'Dictionary' DLL that other programmers can call from thier EXE's. You didn't actually create an EXE, however, the 'other programmers' will call your DLL. In turn, your DLL must get a TASK handle to call MF. Our suggestion: Have the 'other programmers' .EXE call yourDLLInit, which will call mfInit. Return to the other programmers a 'TASK' handle that will be used to call MF functions (through your DLL...). Just make sure you can translate the 'TASK' they pass to the TASK required for MF access.

Q: I am getting a total system lock-up from within Windows. I don't know if it is MF or something else. Help!

A: Well, this can be a problem. However, the following is list of stuff that may cause a total lock or severe GPF:

- Bad 'storage' area passed.
If you pass a 'data' area that can't hold all the data in a record, it may (and probably will) GPF. This is usually the culprit if MF fails.

- Out of stack space. If you're calling from VB, the stack is set to 20k. Since a DLL doesn't have its own stack, it uses your stack! MF will work with VB, however, if you are using C or some other language, be sure to bump the stack size up. We have tried to ensure that recursion never exceeds 10k of stack space, however, keep in mind that if your functions are nested deep, MF may have very little to work with when it gets called. A 20k Stack is usually more than sufficient to prevent this problem. We realize that MS recommends a stack size of 5k for windows applications. Again, we will PROBABLY run just dandy in 5k, however, when you get up around 2-3 million records, there's a lot of recursion taking place.

- Check the return codes! While our examples don't do justice to demonstrating a well behaved application, yours should. (In future releases ours will...) It is unlikely that MF will fail based on a bad parameter OTHER THAN passing a bad 'data' area to be filled. However, users are notorious for moving data files, etc... And, if you assume MF has returned a valid DB handle on an mfOpen... Well, it just won't work.

Q: Is MF a multiuser system?

A: Yes. However, This release does not support an internal locking mechanism. Index's are 'locked' and updated properly in a multi-user environment, so you don't need to 'LOCK' and 'UNLOCK' a record to update it. It is 'implied' by calling 'mfWrite' that a lock and unlock should be performed. If you want to 'lock' records to prevent other users from accessing them, you will need to implement something on your own (for now...). We recommend setting a character field or something of that nature, for now.

In a future release, we will have a C/S lock manager that will handle all of this.

NOTE: Lock and UnLock has been added 11/11/93

Q: What does 'MF' stand for?

A: We haven't decided. We may even change the name. When this project was started, MF stood for 'something', but no one remembers what it stood for anymore. Now, it's just matter of a global replace once we come up with a name.

End of file