

Lockfiles vs. Record Locking

By Kenneth Brody

This topic seems to come up repeatedly over the years, so it seems a natural for this "back to basics" issue of the Journal. (I wrote an article for Smalltalk back in 1992, and even then it was nothing new.) Basically, it can be summarized in a few words: "lockfiles have nothing to do with record locking".

Record locking

When filePro enters update mode on a record, it locks that record to prevent any other filePro program from updating the same record at the same time. This is done by filePro requesting the lock from the operating system, which will either grant the lock (and filePro continues into update mode) or denies the request because another program has already locked that record. Keeping track of who has what locked is handled entirely by the O/S. If a program crashes, all locks held by that program are automatically cleared by the O/S. If the system crashes, all information regarding all locks are lost, and no locks are in place upon reboot.

Note, however, that the Internet has added a layer of confusion to this. If you are connected to a remote computer (via "telnet", for example) and your system crashes while a program you were running on the remote computer has records locked, it is possible for those locks to remain in effect. This is due to the fact that the other system does not know that your computer is no longer connected, and it is holding the connection open. Therefore, your program is still running and still holding the lock. If this happens, you will need to kill the program manually. Otherwise, the lock will remain in effect until the other system shuts down the connection. (And on many systems, this may not be until the system is rebooted. I, personally, have seen such dropped connections remain for several days.)

Lockfiles

Lockfiles, on the other hand, are maintained exclusively by filePro to maintain inter-module locking, when record locking doesn't suffice. For example, if someone is in define files changing the file structure, you need to prevent anyone from accessing the file at all. Conversely, if anyone is accessing the file, you need to prevent someone from going into define files and changing the file structure. This is done by filePro setting flags and counters within the file's lockfile.

Because this is an external file, it is possible for it

to get out of sync. If a program crashes before it has been able to clear its entry from the lockfile, it will remain marked as "in use". Similarly, if the system crashes while any filePro program is running, the lockfiles are not automatically cleared upon reboot. The symptoms of this situation would be getting "not available" errors (filePro error 33) when no one is actually using the file.

filePro used to include an "auto unlock" script on Unix, but there are so many different variations of Unix systems today that there is no single location to put such a script that would work across all platforms. Typically, you would need to place such a script somewhere under the `/etc/rc.d` directory tree. Check with your system administrator for the proper location for your particular system.

Note that the proper way to delete lockfiles is to use the `-l` flag to `ddir`. Unfortunately, there is no way to tell `ddir` to remove all qualified lockfiles for a given file. For an auto-unlock that gets run upon system startup, it's probably acceptable to simply `rm` the files. However, you should never clear them via a `>lockfile` command.

The "-u" flag

One final item that seems to confuse many people is the `-u` flag on `*report`. Just like lockfiles, it too has nothing to do with record locking. Rather, it affects the inter-module locking provided with the lockfile.

By default, any report that is run that has output processing associated with it will not be allowed to run if anyone else is in that file. This is to prevent possibly getting reports that don't accurately reflect the data in the file. (After all, it takes a finite amount of time to run the report, and other people may be changing the data during that time.) If it is acceptable to have data changing while the report is running, then you can pass the `-u` flag to `*report`. This will allow the report to run, despite there being other people in the file. It will not, however, stop the report from performing the record locking needed when updating records.