



Understanding Record Locking

I teach a lot of FileMaker courses and I am always surprised at the tremendous number of experienced developers who don't consider record locking issues in their development. Usually students have heard about the concept but don't know how to replicate a record locking issue, let alone test for a record locking error in a script. Understanding all aspects of record locking is extremely important to producing a professional FileMaker solution.

Let's start by defining record locking. Record locking occurs when two Guests try to access the same record at the same time. In FileMaker 6 and earlier versions, a Guest could take control of a record simply by clicking into a field. Subsequent Guests attempting to access the same record would receive a message that someone else was modifying the record. The record would remain locked until the first user released the lock by exiting all fields. Exiting all fields could occur by typing the Enter key on the keypad, clicking outside the fields, changing records or changing layouts, just to name a few. The only difference with FileMaker 7 and later is record locking doesn't occur until a Guest starts modifying a record. In other words, someone has to start typing into a field on the record.

Once you understand record locking, the next step is to learn how to test for record locking without setting up a Host computer, copying your solution to the Host hard drive, and sharing it via FileMaker Server. This might not seem so difficult until you realize that you have to copy a new version of the solution to the Host each time you want to test a new version of the solution. An easier approach, and the only approach if you only have one computer for development, is to open your solution in regular FileMaker Pro and turn on sharing. You'll find the FileMaker Network menu choice underneath the Sharing sub-menu under the File menu. When the resulting dialog appears, turn on network sharing and make sure the file you are testing allows access by all users.

Once file sharing is turned on, the same computer will act as Host and Guest. Just choose Open Remote from the File menu. In the resulting dialog, locate your computer under local hosts. Open the solution you want to test for record locking and a second occurrence of your file will appear, using the same name in the title appended with "-2" and the name of your host. Now you are ready to test record locking. In either one of the windows, click into a record and begin typing. Without exiting the record, switch to the second window and attempt to modify the same record. You will receive an error message telling you it is being modified.

The standard error message that FileMaker generates when two Guests attempt to manually modify the same record works fine. Only when you incorporate scripting into your solution does record locking require your supervision. Any time a script attempts to modify a field on a record, such as occurs with Set Field, Insert Calculated Result and many other script steps, you need to make sure the current Guest has control of the record or else the record may not be modified. Rather than trying to modify the record and then test for an error, a developer is better off trying to gain control of the record without modifying it using the Open Record/Request script step. For instance, the following script shows the basic construct for a record locking aware script.

```
Set Error Capture [On]
Open Record/Request
If [Get(LastError) = 301]
    Show Custom Dialog ["Record Locked"; "This record is
locked!"]
Else
    #Place any script steps here you want to occur if the record
is open
End If
```

Briefly, the Set Error Capture script step hides any error message dialog that might appear from FileMaker such as the record locking error. The Open Record/Request script attempts to gain control of the record. If the record is locked (being modified by another Guest), an error of 301 will result. If no error occurs, the script will have locked the record for the current Guest and the script can safely modify the record. As an aside, a locked record can be released using the Commit Record/Request script step.

Testing if the current record is locked is simple but the task gets significantly more complex when attempting to modify multiple records. For example, the Replace Field Contents script step or a record looping script, attempt to modify multiple records in succession. If you have a few Guests on the network, chances are the modifications will occur without a hitch. When more users are introduced, the chances that one or more records is locked increases.

So, what can you do in this record locking scenario? There are complicated transaction techniques you can employ but I like to tailor the solution to the need. For instance, let's say you want to mark all the records in the found set so you can search for the marker and reconstitute your found set at a later date. Of course, this could cause record locking on more than one record which you could attempt to track. Instead of writing a value to the records, grab the serial number from each record. Reading a value from a record does not require it to be unlocked. In this way, you can gather all the serial numbers and store them in a return-separated list. To restore your found set, simply use the serial numbers in a multi-key relationship along with the Go to Related Record script step. While this is a brief overview of the technique, you get the idea. For every record locking scenario, there is usually an alternative approach.

- John Mark Osborne, www.databasepros.com