January, 1996

Advisor Answers

Q: I have a FoxPro 2.6 application with an SQL SELECT statement to generate a list of information from a database that is not currently in use. Later on in the application, I USE the same database, and I get "File is in use". I can explicitly select that database and issue "USE" to close it after the SQL call, but since the database name could begin with a digit, I do not always know its alias. Is there a way to get SELECT to clean up after itself? If not, has this been fixed in 3.0? If it has been fixed, it will be a big reason for me to make the switch as soon as possible.

–Rob Richardson, Wickliffe, OH (via Internet)

A: First, let's talk about terminology. What you're referring to as a "database", that is, a DBF file, is more accurately called a "table." A database is a collection of tables and the relationships among them. The Xbase community developed the bad habit of refering to tables as "databases" because of the DBF ("DataBase File") extension. Now that Visual FoxPro gives us the ability to define databases, not just tables, it's important that we learn to use the right terms.

As for your problem, it can be solved. The answer is a little different in Visual FoxPro than in FoxPro 2.x, but the basic idea is the same. You check to see what tables are open before the query, then do it again after the query and close the ones that got opened.

We'll do the Visual FoxPro verson first because it'll give us the basis for the FoxPro 2.x version. It uses the new AUSED() function which fills an array with information about the work areas and tables in use.

```
LOCAL nOldUsedCnt,nNewUsedCnt, nCnt    && use PRIVATE in 2.x
LOCAL ARRAY aOldUsed[1], aNewUsed[1]

nOldUserCnt = AUSED(aOldUsed)

* now run your query

nNewUsedCnt = AUSED(aNewUsed)
FOR nCnt = 1 TO nNewUsedCnt
    * see if it was open before
    IF ASCAN(aOldUsed,aNewUsed[nCnt,2])=0 ;
       AND aNewUsed[nCnt,1]<>"<Name of result cursor>"
       * if not, close it
       USE IN aNewUsed[nCnt,2]
    ENDIF
ENDFOR
```

Notice that, in addition to testing whether the table was open before, we also make sure not to close the cursor that resulted from the query.

The same code will run in FoxPro 2.x, except for a couple of problems. The first is the local declarations. That's easy - change the first to PRIVATE and change the array declarations to use DIMENSION.

The second problem is the big one. FoxPro 2.x has no AUSED() function. Fortunately, it's easy to write our own. Here it is:

```
FUNCTION AUSED
* Return an array containing one row for each workarea in use.
* Put the alias in column 1 and the work area number in column 2.

PARAMETERS aResult
EXTERNAL ARRAY aResult

PRIVATE nCnt,nUseCnt,nMaxAreas

* First, figure out how many work areas
IF _DOS
   IF "(X)"$VERS(1)
      nMaxAreas = 225
   ELSE
      nMaxAreas = 25
   ENDIF
ELSE
   nMaxAreas = 225
ENDIF

* now loop through
nUseCnt = 0
FOR nCnt = 1 TO nMaxAreas
   IF USED(nCnt)
      nUseCnt = nUseCnt + 1
      DIMENSION aResult[nUseCnt,2]
      aResult[nUseCnt,1] = ALIAS(nCnt)
      aResult[nUseCnt,2] = nCnt
   ENDIF
ENDFOR

RETURN nUseCnt
```

Here's the FoxPro 2.x version of the original code. The only other change is that the arrays must exist before the call to AUSED() and you have to remember to pass them by reference.

```
PRIVATE nOldUsedCnt,nNewUsedCnt, nCnt
DIMENSION aOldUsed[1], aNewUsed[1]

nOldUserCnt = AUSED(@aOldUsed)

* now run your query

nNewUsedCnt = AUSED(@aNewUsed)
FOR nCnt = 1 TO nNewUsedCnt
    * see if it was open before
    IF ASCAN(aOldUsed,aNewUsed[nCnt,2])=0 ;
       AND aNewUsed[nCnt,1]<>"<Name of result cursor>"
       * if not, close it
       USE IN aNewUsed[nCnt,2]
    ENDIF
ENDFOR
```

You'll find the 2.x version of AUSED() and both versions of the example program (TestClos.PRG) on this month's Companion Resource Disk.

–Tamar