September, 1999

## Ask Advisor

Q: How can I get a list of all databases that are open?

—Chico (via Advisor.COM)

A: I don't know which version of FoxPro you're working with and the answer to this question is different in Visual FoxPro than in FoxPro 2.x. I'll look at VFP first.

VFP has functions to tell you both which database containers (.DBC) are open and which tables (.DBF) are open, as well as other functions that help you check on the current environment.

ADATABASES() fills an array you pass it with a list of all open databases. The first column of the array contains the database name, while the second column contains the full path to the database. Like the other array functions, it returns the number of items it finds, in this case, the number of open databases. Here's a simple example:

```
* Set up some data to find
CLOSE DATABASES ALL
OPEN DATABASE (_SAMPLES + "\TASTRADE\DATA\TASTRADE")

* Get the list
nOpenDBCs = ADATABASES( aDBCsOpen )
```

This code sets nOpenDBCs to 1 and creates aDBCsOpen as a one row array where aDBCsOpen(1,1) contains "TASTRADE" and aDBCsOpen(1,2) contains the full path to TasTrade. On my machine, that's:

```
G:\MSDN\VS6\98VS\1033\SAMPLES\VFP98\TASTRADE\DATA\TASTRADE.DBC
```

Note that ADATABASES() doesn't pay any attention to data sessions. That's because databases themselves aren't open in just a single data session. Once you open a database, it's available to every data session.

To determine whether a particular database is open, use the DBUsed() function. Pass it the name of a database and it returns a logical value indicating whether the database is open. You might use it to make sure a certain database is open before SETting DATABASE to it, or only open a particular database if it's not already open:

```
IF DBUsed("MyDatabase")
   SET DATABASE TO MyDatabase
ENDIF

IF NOT DBUsed("AnotherDatabase")
   OPEN DATABASE AnotherDatabase
ENDIF
```

The DBC() function provides the full path to the current database. So, on my machine, if TasTrade is the current database:

```
? DBC()
```

displays the path shown above. You can find out just the name of the current database with SET("DATABASE").

Moving on to tables, AUSED() is the ticket in VFP. This function fills an array with the list of tables open in the specified data session (the current session, if no session is specified). The resulting array has two columns, with the alias in the first and the workarea number in the second. The function returns the number of tables found. For the list of tables in the current data session, use:

```
nTablesOpen = AUSED( aTables )
```

To examine a different data session, specify the session number as the second parameter:

```
nTablesOpen = AUSED( aTables, nDataSession )
```

To check on a particular table, try USED(). You can pass either an alias or a workarea number; the function returns a logical value. If you pass an alias, it tells you whether any workarea has a table open with that alias. (In VFP, the search is limited to the current data session.) Passing a workarea number tells you whether there's any table open in the specified workarea. With no parameter, USED() indicates whether there's anything open in the current workarea.

The most common appearance of USED() is in code like this:

```
IF USED("SomeAlias")
   SELECT SomeAlias
ELSE
   SELECT 0
   USE SomeTable AGAIN ALIAS SomeAlias
ENDIF
```

My co-columnist, Christof Lange, comments that he'd write the same code this way, because it's a little shorter without a loss of clarity:

```
IF NOT USED("SomeAlias")
   USE SomeTable AGAIN ALIAS SomeAlias IN 0
ENDIF
SELECT SomeAlias
```

Which approach you use is a matter of personal taste, but do try to be consistent and use the same technique in all your code. Otherwise, those maintaining your code will be stuck trying to figure out why you did one way in some places and the other way elsewhere.

ALIAS() accepts a workarea number and tells you the alias of the table open in that workarea. For example, if the Customer table is open in workarea 2, ALIAS(2) returns "CUSTOMER".

In FoxPro 2.x, you can use ALIAS() and USED() to build your own version of AUSED(). (Actually, you can do it with just ALIAS(), but USED() makes it a little more readable.) Here's a version that first appeared in this column in the January '96 issue. You'll also find it on this month's Professional Resource CD.

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

PARAMETERS aResult
EXTERNAL ARRAY aResult

PRIVATE nCnt, nUseCnt, nMaxAreas

* First, figure out how many workareas
IF _DOS
   IF "(X)"$VERSION(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
```

Calling this function in FoxPro 2.x is slightly different from using the built-in function in Visual FoxPro. The array must exist and you have to precede the array name with "@":

```
DIMENSION aTables[1]
nTablesOpen = AUSED( @aTables )
```

Finally, to round out the set of functions, DBF() accepts an alias or workarea and provides the filename of the table open in that workarea. On my machine, for the TasTrade Customer table, DBF("Customer") returns:

```
G:\MSDN\VS6\98VS\1033\SAMPLES\VFP98\TASTRADE\DATA\CUSTOMER.DBF
```

DBF() honors the FULLPATH setting, so if FULLPATH is OFF, it returns only the drive and table name without the path. So, on my machine:

```
SET FULLPATH OFF
? DBF("Customer")
```

returns:

```
G:CUSTOMER.DBF
```

Taken together, these functions provide plenty of information about what's going on data-wise in your application.

—Tamar