June, 1998

Advisor Answers

Visual FoxPro 5.0 and 3.0 and FoxPro 2.x

Q: How do I tell Visual FoxPro to keep .DBF files in 2.6 format? I am trying to migrate to Visual FoxPro. In doing so I will write new applications in VFP, but we have many legacy apps (most written in FoxPro/DOS) in which the .DBF's are FoxPro 2.6. How do I make sure those applications can still access the data when the VFP applications are also using it?

–Patrick Dohogne (via the Internet)

A: As more and more people are in your situation, with new development in VFP, but legacy applications in FoxPro 2.x, this is a timely question. The good news is that, actually, it's quite easy to ensure that VFP doesn't change the structure of your tables.

The biggest issue is that FoxPro 2.x tables cannot be added to a database. As soon as you issue ADD TABLE or use the interface to add an existing table to a .DBC, its format is changed to VFP format.

Does this mean that you can't have any of the advantages of databases with your legacy tables? Fortunately, no. You can have the best of both worlds by keeping the tables themselves free, but putting views of those tables into your database. The views can take advantage of many database features.

The other big piece of keeping databases in FoxPro 2.x format is to make sure you don't ask for any features that are VFP-only. Adding a field of one of the VFP data types (like DateTime) to a table is a sure way to convert it to VFP format. So is adding support for null values.

If you COPY a FoxPro 2.x table and don't specify otherwise, the new table is a VFP table. However, the original table is unchanged.

I was very surprised in my testing to find that some of the operations that rewrite the table *don't* change it to a VFP table. For example, PACKing a 2.x table leaves it in FoxPro 2.x format. You can use ALTER TABLE or the Table Designer and, as long as you don't add one of the new field types, or ask for nulls, or another VFP-only feature , the table stays in FoxPro 2.x format. I was even able to add a candidate index through the Table Designer without changing the table's format. (I don't recommend doing this in an application, since there's nothing on the FoxPro 2.x side to keep the values in that index unique, and nothing in VFP that rechecks them when you open the table there.)

You can check the format of a table at any time using the SYS(2029) function. It returns a number indicating the type of table it found. VFP tables return 48, while FoxPro 2.x tables return 245. (I have no idea why these particular numbers are used, but they're all documented in Help and they correspond to the first byte of the file, which indicates its type.)

If you accidentally change a table to VFP format, it's easy to restore it. Just use the TYPE FOX2X clause of COPY TO, then rename the new file:

```
USE My2xTable&& accidentally converted to VFP
COPY TO MyFixTable TYPE FOX2X
USE
RENAME MyFixTable.DBF TO My2xTable.DBF
* RENAME FPT and CDX, too, if you need to
```

Obviously, in an application, you'd want to add some error checking to that code, but that's the basic structure.

You could even set your application up so that, at the end of each VFP session, it checks the tables that aren't supposed to be converted, and if they've been changed, restores them to 2.x format.

–Tamar