

May, 1997

Advisor Answers

Visual FoxPro

Q: Is there a way to address a table in a data session other than the currently selected one? Often I have found I wanted to get a field value such as names.firstname when it was in another data session.

–Steve Finegold (via the Internet)

A: There's no way to "alias" a field reference by data session so that you could say something like "get names.firstname from the copy of names that's open in data session 3." That's because doing such a thing defeats the whole purpose of using private data sessions.

FoxPro has allowed us to open multiple copies of a table for a long time - the AGAIN clause was added to USE in FoxPro 2.0. But before Visual FoxPro, opening multiple copies of a form was really tricky because each copy had to either assign different aliases to the tables it used or you had to keep track of the status of each table each time you moved from one copy of the form to another. In fact, the same issues arose any time you wanted to open multiple forms if they used any of the same tables.

The data session mechanism in Visual FoxPro solves these problems. Each data session stands alone: you can use the same aliases in each and each one keeps track of the record pointer, the index order and other details of the table. (Actually, the separation isn't quite complete. If two data sessions SET ORDER TO the same index tag, they both have to be either ascending or descending. You can't SET ORDER TO MyTag ASCENDING in one data session and SET ORDER TO MyTag DESCENDING in another. Both sessions use the last order set. In VFP3, but not VFP5, a table that's opened read-only in one data session is opened read-only in all subsequent sessions, regardless of the USE command or settings used. I consider both of these behaviors bugs.)

Notwithstanding the problem with index order, the whole idea of private data sessions is that you can work on a table or tables in one session without being concerned about what's happening in other data sessions. If you could access fields across data sessions, you'd lose that isolation.

The simplest and generally best way to give a form access to a table that's already open is to ignore the fact that the table is already open and just add it to the form's data session. This is a low-cost move, since once a table has been opened by one data session, any other data sessions that use it in effect USE it AGAIN, that is, you only have the overhead of actually opening the table on disk once. Including the table in the form's private data session is the right answer if you want to do something unrelated to the other form(s) in which the table is being used.

What if you need to get a look at the current record of a table open in another data session? This situation is most likely to occur when an action on a form opens a modal form. For example, clicking a button on an overview form might open a form showing details of the current record.

There are two ways to get a modal form to participate in another form's data session; each has some problems. First, a modal form can share the data session of the form that calls it. Just leave the DataSession property of the modal form set to 1-Default and when the form opens, it jumps right into the calling form's data session. Don't put anything in the data environment for this form since it gets access to all the tables of the calling form.

You can use the same approach to run a report of the current record from a form. Don't check the Private Data Session menu item for the report and put nothing in the Data Environment. Then, in the form, when you want to run the report, issue REPORT FORM MyReport NEXT 1.

The problem with this approach is subtle. It's best seen by opening the Data Session window (known as the View window in VFP3) while testing. The private data session of the original form is known by the name of that form. After calling and returning from the modal form, that data session now indicates that it's "Unknown." It appears to work correctly and closes when it should, but the change in identity to "Unknown" is a cause for concern. In addition, Contributing Editor Drew Speedie ran into memory problems when testing this technique (although several other people indicate that they're sharing data sessions this way in their applications without problems.)

The alternative approach is to use the SET DATASESSION command to change the data session of the modal form. The problem here is that you can only do so safely if none of the controls on the modal form are bound. See Drew Speedie's article in this issue for details on making this approach work.

What if you want one form to be able to see the tables open in another and it's not a case of a modal form being called? In that case, you may want to rethink your design. If the user starts these forms independently, why is there communication between them? The user probably sees them as separate tasks and sending information from one to the other could cause confusion. If the tasks are related, then a single form or a formset with a single data session may be a better approach.

-Tamar