January, 2003

Advisor Answers

Docking IDE windows

VFP 8

Q: I like the docked windows in VFP 7, but I could never get them to behave between sessions. Does VFP 8 improve this behavior?

–Name withheld by request

A: Back in the December '01 issue, I wrote about the new docking features of VFP 7, and concluded that they weren't quite ready for prime-time. Fortunately, VFP 8 includes major improvement in docking of IDE windows.

First, VFP 8 remembers the docking status and position of the windows. So, if you have several IDE windows docked when you close VFP, the next time you open it, those windows open up just as you left them. This change alone means that you may never need to dig any deeper into the new capabilities.

However, if you want absolute control over the docking of IDE windows, you'll want to explore the new DOCK WINDOW command. This command lets you dock and undock IDE windows and toolbars. You can dock things to the borders of the VFP desktop or to other IDE windows. Once windows are docked together, you can dock the whole group to the VFP borders. The command provides for both link docking and tab docking.

The first question is which windows are affected; the list is longer than you might think. All of VFP's built-in toolbars can be docked to the sides of the VFP window. They can't be link docked or tab docked, however.

The Debugger windows can be docked. When the Debugger is set to the FoxPro frame, Debugger windows can be tab docked and link docked with any other dockable windows. In the Debug frame, they can be docked to the borders of that frame, but not link docked or tab docked.

What you might consider the three principle IDE windows, the Command window, the Data Session window, and the Property sheet, are all dockable, both to the borders of the VFP desktop, and using tab docking and link docking.

Finally, the Document View window can be docked, either at the borders or with other windows through tab docking and link docking.

So, how do you dock windows? You can do it either interactively or with the DOCK WINDOW command. I discussed interactive docking of windows in the 12/01 article, so I won't go over that again. Instead, let's look at DOCK WINDOW.

The syntax for this command is:

```
DOCK WINDOW WindowName POSITION nPosition
      [WINDOW TargetWindowName]
    | WINDOW TargetWindowName
```

The syntax is a little hard to follow, but essentially you have three choices. First, you can dock a window to or undock a window from a VFP border using:

```
DOCK WINDOW WindowName POSITION nPosition
```

Second, you can tab dock a window to another window using:

```
DOCK WINDOW WindowName WINDOW TargetWindowName
```

Finally, you can either link dock or tab dock a window to another using:

```
DOCK WINDOW WindowName POSITION nPosition ;
     WINDOW TargetWindowName
```

The POSITION value determines where a window is docked and, in some cases, which kind of docking is used. The values (shown in Table 1) can be divided into three groups.

Table 1 Where to dock–The value of nPosition determines whether a window is docked or undocked, where a window is docked, and in some cases, what type of docking is used.

nPosition	**Meaning**
0	If WINDOW clause is omitted, dock at top of VFP or Debug frame. If WINDOW clause is included, link dock with the specified target window, putting this window at the top.

nPosition	Meaning
1	If WINDOW clause is omitted, dock at left of VFP or Debug frame. If WINDOW clause is included, link dock with the specified target window, putting this window at the left.
2	If WINDOW clause is omitted, dock at right of VFP or Debug frame. If WINDOW clause is included, link dock with the specified target window, putting this window at the right.
3	If WINDOW clause is omitted, dock at bottom of VFP or Debug frame. If WINDOW clause is included, link dock with the specified target window, putting this window at the bottom.
4	Tab dock this window with the specified target window. If the WINDOW clause is omitted, an error is generated.
-1	Undock this window. If the window is part of a group of windows docked together, remove it from the group. If the window is docked to a border, undock it from that border. If the window is part of a group that's docked at a border, remove it from the group and undock it. If the WINDOW clause is included, an error is generated.
-2	If the window is part of a group of windows (tab docked or link docked together) that's docked to the VFP border, undock the group as a whole, leaving the windows tab docked or link docked as they are. If the window is not part of a group, but is docked at the VFP or Debug frame border, undock it. If the window is not docked, the command is ignored. If the WINDOW clause is included, an error is generated.
-3	If the window is part of a tab docked group that's link docked to additional windows, remove the whole tab docked group from the link docked group. This occurs whether the containing link docked group is docked to a border or not. In all other cases, behave as for nPosition = 2. If the WINDOW clause is included, an error is generated.

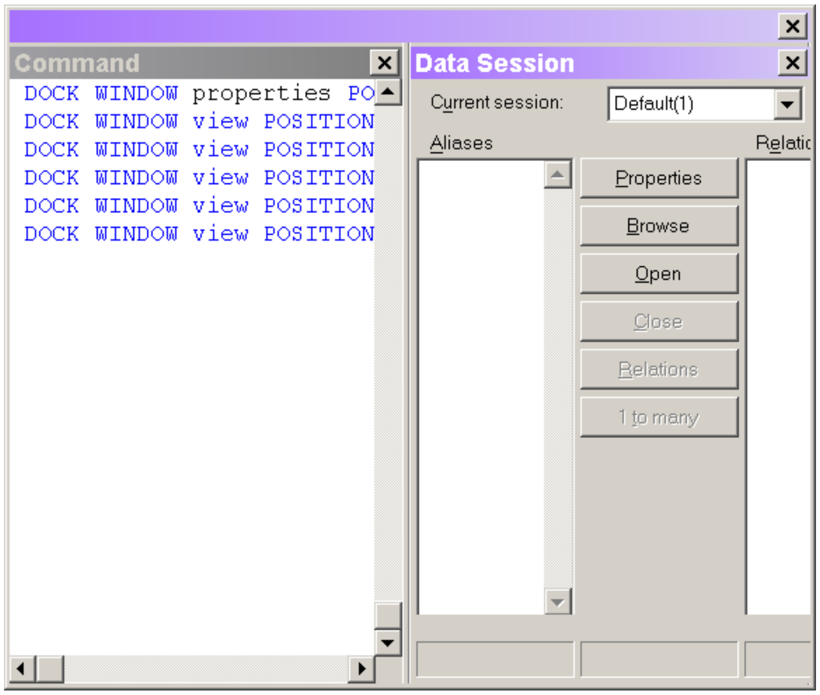
The first group, 0 through 3, lets you dock along borders. If the WINDOW TargetWindowName clause is omitted, the specified window is docked at a VFP border (or, for debugger windows in the Debug frame, at the Debug frame border). For example, issuing:

```
DOCK WINDOW View POSITION 2
```

docks the Data Session window (known as "View" for historical reasons) to the right side of the VFP desktop. If the WINDOW clause is included, the specified window is link docked with the target window. So, this command:

```
DOCK WINDOW View POSITION 2 WINDOW Command
```

link docks the Data Session and Command windows with the Data Session window on the right, as in Figure 1.



```
DOCK WINDOW properties PO
DOCK WINDOW view POSITION
DOCK WINDOW view POSITION
DOCK WINDOW view POSITION
DOCK WINDOW view POSITION
DOCK WINDOW view POSITION
```

Figure 1 Link docking–In VFP 8, you can link dock windows using the DOCK WINDOW command.

The second "group" of values has only one member. When nPosition is 4, the specified windows are tab docked. For example, issuing:

```
DOCK WINDOW View POSITION 4 WINDOW Command
```

tab docks the Data Session and Command windows, with the result shown in Figure 2

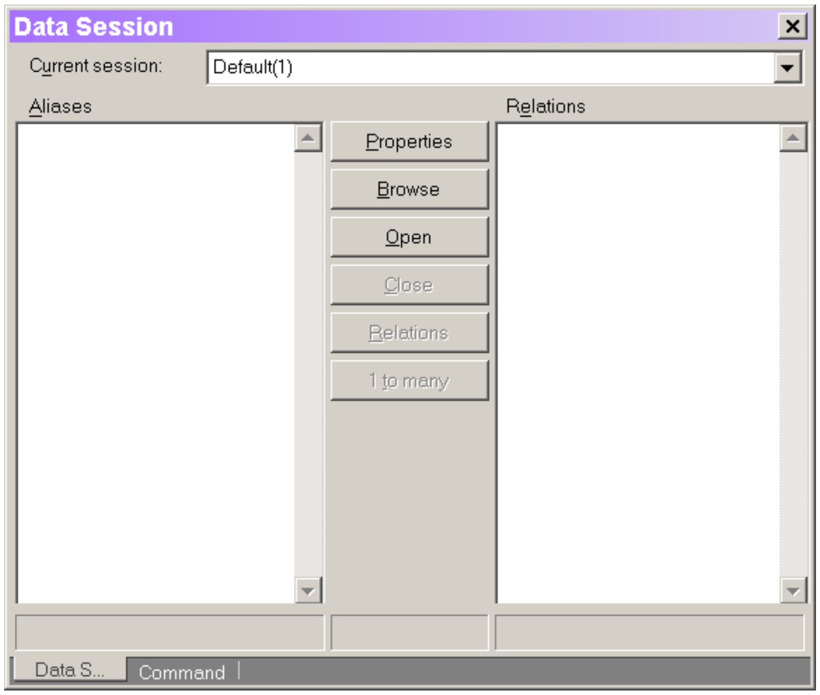
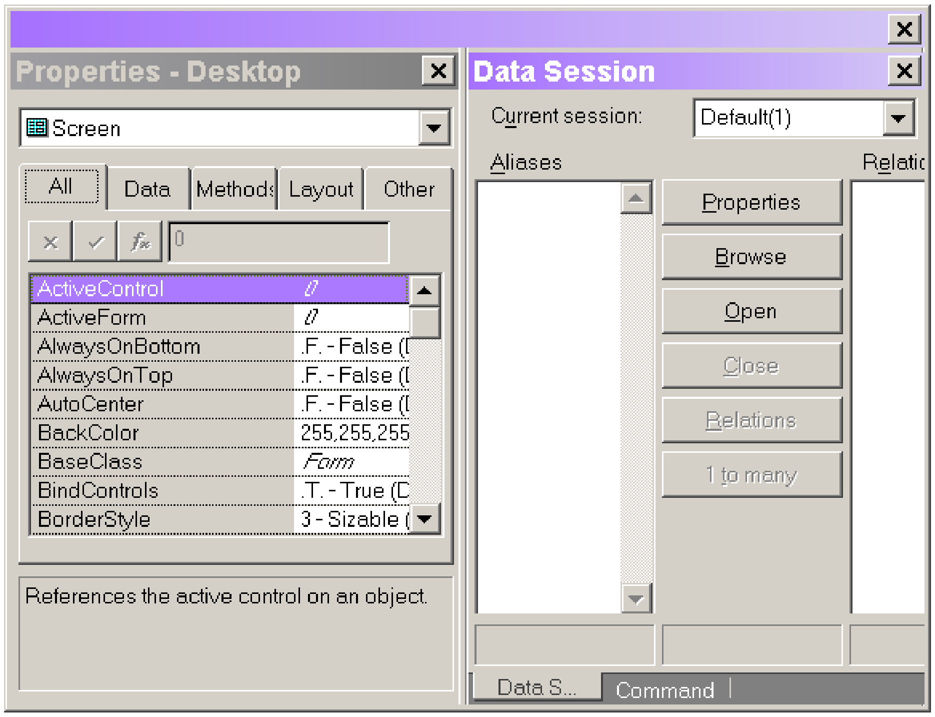


Figure 2 Tab docking–Specifying 4 for nPosition results in tab docking.

Finally, the third group of values for nPosition deals with undocking of windows. Passing –1 undocks the window, no matter where it's docked or to what. But sometimes, you want to be more subtle than that. If you have several windows docked together, and that group is docked to a border, you may want to undock the group as whole rather than the individual windows; use –2 for nPosition, in that case. It can be even more complicated. If you have some windows tab docked, and that group of tab docked windows is link docked to other windows (as in Figure 3), you need a way to extract the whole group of tab docked windows from the link docked group; in that case, specify –3 for nPosition.



ActiveControl	0
ActiveForm	0
AlwaysOnBottom	.F. - False (
AlwaysOnTop	.F. - False (
AutoCenter	.F. - False (
BackColor	255,255,255
BaseClass	Form
BindControls	.T. - True (D
BorderStyle	3 - Sizable (

References the active control on an object.

Figure 3 Mixed docking–You can link dock a tab docked group with another window, or even with another tab docked group. To take groups like this apart programmatically, you need the –2 and –3 values for the POSITION clause.

A few more notes. When you issue DOCK WINDOW for any window, its status is changed to dockable. That is, you don't need to issue WDOCKABLE() first to make a window dockable.

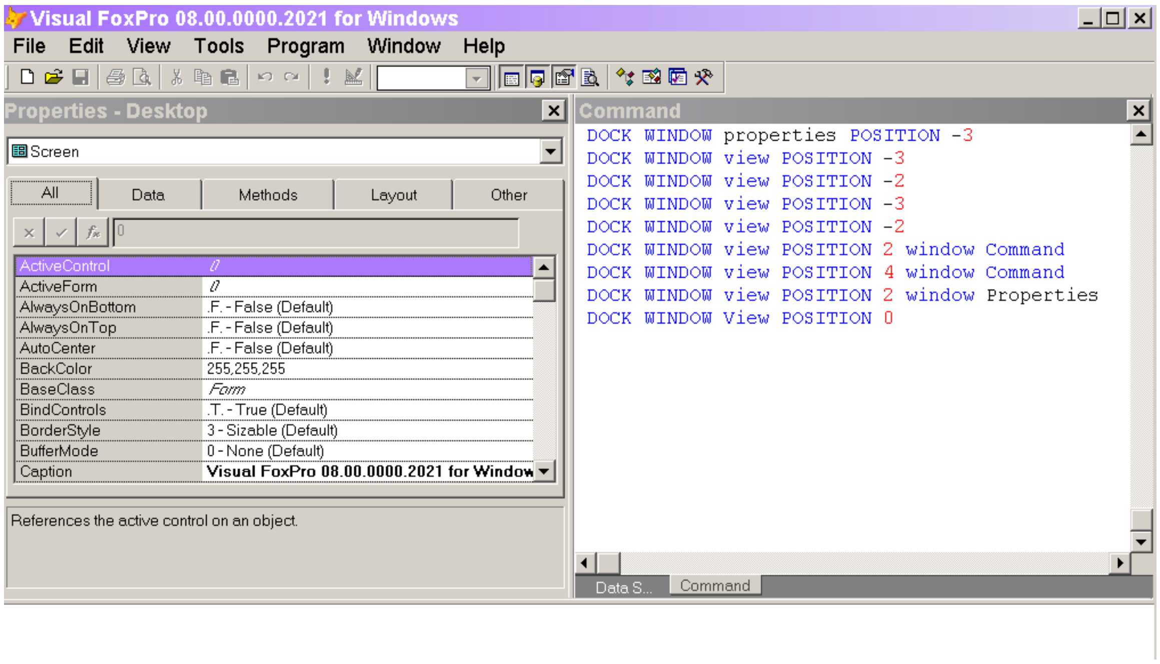
Once you've tab docked or link docked a group of windows, you can dock the whole group by issuing DOCK WINDOW for any member. So, to create the configuration in Figure 3, you can use these two commands:

```
DOCK WINDOW View POSITION 4 WINDOW Command
DOCK WINDOW View POSITION 2 WINDOW Properties
```

To then dock the whole shebang at the top, use:

```
DOCK WINDOW View POSITION 0
```

Another nice change in VFP 8 (shown in Figure 4) is that when you dock a link docked group to a VFP border, the outer window disappears.


Visual FoxPro 08.00.0000.2021 for Windows
File Edit View Tools Program Window Help
Properties - Desktop
Screen
All Data Methods Layout Other
Command
Data S... Command

ActiveControl	0
ActiveForm	0
AlwaysOnBottom	.F. - False (Default)
AlwaysOnTop	.F. - False (Default)
AutoCenter	.F. - False (Default)
BackColor	255,255,255
BaseClass	Form
BindControls	.T. - True (Default)
BorderStyle	3 - Sizable (Default)
BufferMode	0 - None (Default)
Caption	Visual FoxPro 08.00.0000.2021 for Window

References the active control on an object.

```
DOCK WINDOW properties POSITION -3
DOCK WINDOW view POSITION -3
DOCK WINDOW view POSITION -2
DOCK WINDOW view POSITION -3
DOCK WINDOW view POSITION -2
DOCK WINDOW view POSITION 2 window Command
DOCK WINDOW view POSITION 4 window Command
DOCK WINDOW view POSITION 2 window Properties
DOCK WINDOW View POSITION 0
```

Figure 4 Docking link docked groups to borders–In VFP 8, the outer window containing a link docked group disappears when you dock that group to one of VFP's borders.

Finally, since you can now manipulate docking programmatically, you need a way to check the current status. The new ADOCKSTATE() function fills an array with information about docked windows.

Now that we have real control over docking and undocking, and persistence between sessions, it's worth spending some time experimenting with different configurations to figure out what set-up is most productive for you.

–Tamar