October, 1997

Advisor Answers

Changing Menu Fonts

FoxPro 2.x and Visual FoxPro

Q: We are developing an Application in VFP 5.0. We have designed our forms in 800 x 600 and our application is supposed to be run at that resolution. We have designed the menu using the Menu Designer.

Our specific problem is we want to show the menu options in a bigger size. The menu is picking up the default Windows 95 resolution setting and there is no option in the Menu Designer to control this. We know we can control the menu font, size and style programmatically, but is there a way to do so in the Menu Designer?

–Subrata Datta (via Internet)

A: With VFP's Menu Designer alone, there's no direct way to control the font used for the menu. However, the public domain tool, GENMENUX, written by Andrew Ross MacNeill, includes a directive (*:FONT) that lets you specify the name, size, and style for the menu font.

GENMENUX is a wrapper for FoxPro's GENMENU program. (GENMENU turns .MNX/.MNT files into .MPR menu programs.) GENMENUX is similar to Ken Levy's GENSCRNX which was an invaluable tool for FoxPro 2.x. Unlike GENSCRNX, though, GENMENUX is still useful with VFP because menu generation hasn't changed much. GENMENUX includes a number of directives that let you enhance the menus FoxPro generates, including making some menu items optional, changing the menu color and much more. It also includes hooks for directives that can perform much larger changes to menus. You'll find GENMENUX and its documentation on this month's Companion Resource Disk. (Also, see the April '94 issue for an introduction to GENMENUX.)

Now that I've told you how to get what you want, I have to add that I think, in most cases, changing the menu font or size is a bad idea. In the Windows environment, fonts and colors and so forth are supposed to be under the user's control. In Windows 95 and Windows NT, it's particularly easy for the user to change such things. (Right-click on the desktop, choose Properties, then choose the Appearance page. To change the menu font, choose Menu in the Item dropdown and select the font, size, style and color you want. See Figure 1.) In fact, it's possible to have multiple sets of choices saved as schemes and choose among them at different times. For example, on my notebook computer which I often use to give presentations, I have a saved scheme called "Session Scheme" which sets the menu and message fonts to 12 points so that people in the audience can see them.

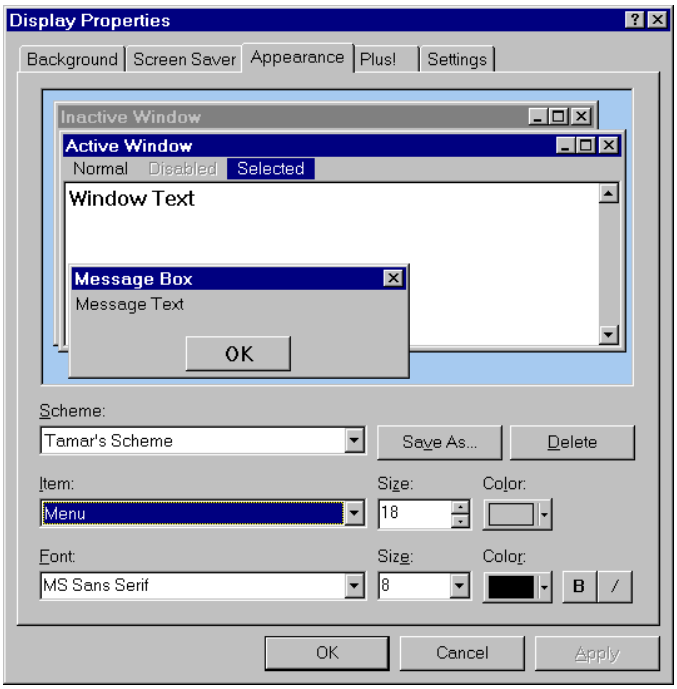

Figure 1. Changing Menu Settings - The Appearance page of the Control Panel's Properties dialog lets you change many aspects of your system's visual setup.

Although many users are unaware they can change these settings, those who know how to change them will not be happy with an application that overrides their choices. I would be quite annoyed by an application that didn't pay attention when I change to my session scheme. I imagine that a number of people with vision problems also take advantage of the ability to customize the appearance of their work area. They'd probably go well beyond annoyance at an application that didn't listen.

If your users complain that the menu isn't readable, teach them how to customize their settings rather than just taking over and ignoring them. Most users will be delighted to learn how much control they have over their machine's appearance. You could even include the option to launch the control panel options directly from your application. The command for both Windows 95 and Windows NT 4.0 is:

```
RUN /n control.exe desk.cpl
```

The lesson here is one I've been preaching for quite some time (most recently in the June '97 Editor's View) - it's the user's computer, not the programmer's. Your applications need to respect the user's settings and live with them. Finally, just because you can do something doesn't mean you should.

–Tamar

Putting Grids in Grids

Visual FoxPro

Q: I am trying to create a grid that contains several grids inside for the purpose of showing several filtered views of the same table. I don't have enough screen real estate to put all the grids on the same form. I also want to be able to show all of them at once, not on separate pages of a pageframe. I can create the grid with no problems but I

cannot get all the columns to display the grids at the same time. They only display when they have focus.

Is there a property or other way of displaying all the column's CurrentControls at the same time?

–Charles Obadia (via Internet)

A: Although your situation (every column of a grid containing another grid) is a little extreme, the problem you face is a common one. In order to speed up display and make things look less cluttered, the default behavior of grids is to show the CurrentControl only for the current row. For grids embedded in grids, it appears the economy goes a little farther - the grid shows only when the cell containing it has focus. All other cells show a default textbox over which you have little control in VFP 3 and somewhat more control in VFP 5. The default textbox that appears when the cell doesn't have focus is *not* the same as the textbox that is initially assigned to each column as its CurrentControl (call this one the "automatic textbox" for lack of a better term).

To force a column to show its CurrentControl in every row whether or not the cell has focus, change the column's Sparse property to .F. This should give you what you need to display multiple grids simultaneously. If you wish, rather than having to set each column individually, you can use the grid's SetAll method to change the property for every column. In the grid's Init, issue:

```
THIS.SetAll("Sparse",.F.)
```

If you'd rather take care of it at design-time, you can write a little builder to issue the SetAll command or just do it yourself from the comment window. First, you need to get a reference to the grid. Click on it, then in the command window (or your builder program), call the ASELOBJ() function:

```
ASELOBJ(aGrid)
```

The function creates the array aGrid and populates it with references to the objects currently selected. Since only the grid is selected at this point, aGrid has a single row. To set the Sparse property of all columns to .F., issue the command:

```
aGrid[1].SetAll("Sparse",".F.")
```

This kind of manipulation of objects from the command window or code at design-time is one of the most powerful features of VFP for developers.

When you want to put controls other than grids in the columns of a grid, it may actually be handy to leave Sparse at its default value of .T. for many columns. A grid full of dropdowns, spinners, and other controls can be overwhelming.

In VFP3, leaving Sparse set to .T. is often frustrating because there's no mechanism for controlling the appearance of the default textbox. However, VFP 5 adds Format and InputMask properties to columns. These properties apply to the default textbox. If a textbox is present and its own Format and InputMask aren't set, the columns' properties apply to it as well. The reverse is not true - setting Format and InputMask for any textbox contained in the column has no effect on the default textbox.

Format and InputMask operate as a set here - if either Format or InputMask is set for the textbox contained in the column, it ignores the column's properties.

By combining the Sparse, Format and InputMask properties, you can exercise a great deal of control over the appearance of a grid, offering your users whatever seems most appropriate for their situation.

–Tamar