

Colorize Your Editor with XILIGHTS

By Whil Hentzen

As big as Microsoft is, even they have limited resources. As a result, there were a few things that didn't make it into the first release of VFP. Outer joins, a more robust debugger, object oriented menus, and... an editor with color coding. While we'll be waiting a little while for some of these features, you can have color coding in your editor now, with Xitech's XILIGHTS library. And possibly the best part of XILIGHTS is that it works both with FoxPro 2.6/Windows and Visual FoxPro!

XILIGHTS comes packaged in two FLLs, one for each platform, and a DLL for the VFP version that goes into the WINDOWS\SYSTEM directory. In either case, simply place the FLL in your FoxPro path and issue the command

```
set library to XILIGHTS
```

From now on, the various components of an editing window will be shown in color. The default colors are as follows:

Text: Black
Keywords (commands and functions): Blue
Strings: Green
Numeric values: Aqua
Constructs (DO WHILE, SCAN): Light Purple
Comments: Mustard
Directives: Dark Purple
System Memvars: Red

These colors are used in regular editing windows of .PRG files as well as the snippet editing windows in FoxPro 2.6 and code windows in VFP.

Customizing XILIGHTS

XILIGHTS looks for a text file called XILIGHTS.INI in your WINDOWS directory. This file controls the color scheme used for each component in the editing window. You can manually modify this file and specify what color should be used for each component. You can also use the SETCOLOR function (part of the XILIGHTS library) to set the color of a specific component. The first parameter to SETCOLOR is the component and the second is the RGB value that the component should be set to. For example, the commands

```
=setcolor(2,rgb(192,0,0))  
=setcolor(6,rgb(0,0,192))
```

will turn keywords to red and comments to blue, respectively.

The 2.6 version of XILIGHTS also has a RGB() function that correspondes to Visual FoxPro's RGB() function for easier specification of a specific color.

Caveats

First, this is shareware, and the folks at Xitech have decided to incorporate a splash screen that appears every few minutes that reminds you to register. Yeah, it's annoying, but it doesn't prevent you from realistically evaluating the product. Second, in FoxPro 2.6, XILIGHTS expects the name of FoxPro's main window. If you've changed the name of the window, either via your CONFIG.FP file or with the used the MODIFY WINDOW SCREEN command, XILIGHTS will appear not to work.

You can use the FOXTITLE function to identify the name of the main window. For example, I typically change the title of the window to something like

```
"We're developing in FoxPro 2.6a!"
```

so that I can keep straight exactly what I'm using. As a result, XILIGHTS doesn't recognize FoxPro. After installing XILIGHTS, the command

```
=FOXTITLE( 'We're developing in FoxPro 2.6a!' )
```

ensures that XILIGHTS recognizes the window and functions correctly.

Where to find XILIGHTS

The file XLGHT105.ZIP contains both 2.6 and 3.0 versions of XILIGHT. It can be found on this month's Companion Disk as well as in Library 7 of VFOX. Registration is about \$45 (US) and can be made through CompuServe's SWREG mechanism. Details are contained in the documentation that comes with XILIGHT.

Book of the Month -----

Rounding out the series of three Microsoft Press books on software development is Steve Maguire's Debugging the Development Process (ISBN 1-55615-650-2). The title says "Debugging" but the keywords are "Development Process." In Writing Solid Code, Maguire focused on the bug hunting process and programming techniques aimed at their prevention; in this one, he takes a step back to look at the management process of software development. He looks at the ultimate goal: getting quality products out the door on time and within budget - and then analyzes the reasons why this rarely happens. Part of the problem, of course, is bugs, but just as large a part of the problem is the rest of the stuff that goes into software development.

A recurring theme is that many programmers spend a lot of time working on tasks that shouldn't be done. For example, he stresses that programmers should spend their time programming, not attending meetings, preparing for meetings, writing up followups on meetings, and other activities. Another example of tasks that shouldn't be worked on is the infamous 'feature creep' - that syndrome of additional features being added to the project, without a corresponding adjustment in the schedule or other resources. Of course, it's easy to say "Well, don't allow new features to be added." but quite another to put this into practice when the person making the request is your boss, or his boss. Maguire makes a number of practical suggestions and strategies for dealing with this case in the real world.

Another piece of advice that he comes back to several times is that delivering high quality products on time doesn't require super-human efforts and 80 hour weeks as the norm - but careful attention to certain details and keeping one's eye on the ball is all that's necessary. He provides a number of examples of how he's brought out-of-control projects back in line, and without having to work miracles. Every experienced developer will find his examples engaging and true-to-life, and put his solutions to work immediately.