

## Session HENxx

# What do you get when you cross a Fox with a Penguin?

## Part II - The Technical Issues

Whil Hentzen

Hentzenwerke Corporation

### Abstract

When you are asked to connect your VFP app to a Linux back end this year – and you will be – you can either offer them a solution or a referral to someone else who had the foresight to prepare in advance and now knows something you don't.

In order for you to be able to work with a Linux back end, you're going to need to know something about how Linux works, and the best way involves a two step process. First, plunk down a Linux workstation on your desk next to your Windows machine and develop some experience with the new OS. Second, once you have a basic level of comfort with Linux, gained through your experience on a workstation, leverage that knowledge and learn to connect to a Linux server from your Windows machine.

This session shows you both of these processes. What to expect when you go about setting up your Linux workstation, how to set it up, how to connect to your Windows network, how to fit VFP into the mix, and even how you might use it to replace your Windows workstation in some cases. And then how to connect to an existing Linux server, running MySQL or another back end, and then get your VFP apps talking to that back end data.

This session is the meat and potatoes sequel to “Expanding Your VFP Skillset with Linux”. You'll learn the nuts and bolts “how to” procedures to get up to speed with a Linux workstation and connecting to a Linux server.

Note: All attendees will receive a free 3 CD set of Red Hat 8.0 CDs that include workstation and server versions of Linux, OpenOffice.org (the Microsoft Office-compatible office suite) and Evolution, the high-end Linux email client.

### The agenda

The following topics will be covered in this paper. As of this writing, the release version of Visual FoxPro 8.0 was not available, so details in some sections are sketchy. Refer to the conference updates for an updated version of this paper.

- Installing Linux
- Top Ten Conceptual Hurdles with Linux
- How Fox and Linux Can Work Together
- Installing Wine, Fox, MySQL

- Using Fox with Native Data
- Using Fox with MySQL
- Resources

## **Installing Linux**

### ***Choosing a distribution***

With over 100 distributions available for Linux, the prospect of picking just one may seem daunting. However, there are just a few that you have to worry about. The three major distributions as of early 2003 are Red Hat, SuSE, and Mandrake. They all come with an easy to use installer and oodles of applications and utilities.

I chose Red Hat because it's got share of mind and works as well as any of the others. Since you can download all three, if you've got the time and interest, why not get a copy of each of them and try them all out to see what you think?

### ***Bootable CDs***

The major distributions all come with bootable CDs if your machine supports the feature. If not, consider picking up a new drive for \$50 from your favorite mail order place. If that's too much trouble, check out the README file on the first CD to find out how to create a bootable floppy disk. The concept is the same as it was with Windows NT and works a lot easier since you only need one bootable floppy.

### ***Choosing dual boot or sole OS***

The next thing to consider is whether you want to install Linux on a machine that also has Windows on it, or if you want to run Linux on its own machine.

If you install a sole OS configuration, you reduce the number of potential problems since you won't have to worry about making sure the right boot loader is set up properly and all that. However, you'll need a separate machine, and that may dissuade you from going this route.

If you use a dual boot configuration, you can install Linux on unused disk space on your machine and then you'll get a OS chooser each time you start up your machine.

The major distributions all have a step in the install process that allows you to choose which way you want to go as part of the regular installation routine.

### ***Choosing a machine***

If you are going to go the dual boot route, you've probably already gotten a machine picked out. If, on the other hand, you want to do a sole OS installation, you probably have an old machine gathering dust in the corner, and may wonder (1) if it's powerful enough to run Linux, and (2) if Linux will recognize the hardware.

I've installed Linux on a variety of desktop machines starting at a 500 Mhz Pentium 3 with 128 MB of memory, and the performance has been fine. 256 MB is better, as it always is, but you definitely don't need a brand new or nearly new

machine. On the other hand, despite what some diehards tell you, that 486 with 32 MB of RAM may not have the horsepower unless you want to use your Linux box for very specialized work, like a printer server.

### ***Type of installation***

The major distributions all come with the ability to install Linux as a desktop or a server from the same set of Cds. Red Hat's installation, in particular, allows you to choose between desktop, workstation, server, and notebook configurations. The difference between these is what applications and services are automatically included during the installation process.

The Desktop configuration includes your choice of GUI, application software like office suites, email, and browsers, and games.

The Workstation configuration includes a variety of programming tools.

The Server configuration includes a Web server, an Email server, an FTP server, and associated applications.

The Notebook configuration customizes a Desktop configuration to account for specific needs on notebook.

With any of these, you can modify the choices before continuing with the installation. For example, you could choose to have an office suite or a Web server installed as part of your Workstation configuration.

### ***This ain't your papa's Linux install***

Common knowledge says that installing Linux is difficult. You have to know how to compile your kernel, handle unresolved dependencies between packages, be able to identify hardware yourself, and so on.

Nonsense.

That used to be true a couple of years ago, and may still be for some specific distributions, but the majors all have installation routines that are just as easy to use as the current Windows installation routines. I've installed Red Hat 8.0 on a variety of machines, ranging from a 6 year old Micron tower with 128 MB of RAM and an ancient 17" CRT monitor and video card to a brand new Thinkpad notebook hooked up to a state of the art flat panel screen. Each time I've been able to hit the "Next" key about a dozen times and then sit back, waiting for files to copy.

No compiling, no missing dependencies, no mysterious hardware glitches.

This isn't true in every single case, of course. The biggest issue you may run into is an odd piece of hardware - an unusual scanner or a non-standard wireless adapter. But if Linux doesn't find your hardware automatically, it'll display a hardware chooser where you can look for the closest match.

And if you switch hardware down the road, you can use Kudzu, the hardware detection tool that comes with Linux, to make switches as necessary.

### ***Configuration files - where's the Registry?***

Linux does not have a centralized storage location for all configuration and other

installation information. Instead, all configuration data, both for hardware as well as for software, is stored in a set of text files that can easily be edited with any text editor.

As a result, if something goes wrong, you don't have to resort to 'registry hacks' or reinstalling the entire operating system because you don't know what happened. Examining the system logs will point you in the right direction, and then you can edit the appropriate config file to fix the problem.

Don't know what config file to work on? There are hundreds of online support groups, from online forums to mailing lists for any imaginable combination. I subscribe to a number of user group lists as well as a Thinkpad-Linux specific one - and some application specific lists as well. Don't worry if the problem is with your GUI - you can boot the system to a command line interface if necessary and edit the config file in question that way.

## **Linux Updates**

Just because this is Linux doesn't mean that you won't have to update your system. The major distributions have online update mechanisms similar to Windows Update that inventory your system and allow you to download and install updates as you desire. And you aren't forced to agree to a huge EULA with who knows what buried in the fine print - in fact, with Red Hat update, there's not a EULA at all. It's all open source, after all.

## **Linux - Adding and Removing Programs**

Unless you're a very boring computer user, you're going to want to change what applications you've got on your machine. There are two general categories of programs.

First, there are those programs and utilities that come with your distribution, and that are thus already part of the Cds, but that you didn't already install (or that you did install, but want to get rid of.) In this situation, you simply run the appropriate tool for your distribution (with Red Hat, it's found under System Settings | Packages) and you'll get a screen that looks much like the one where you configured your distribution during initial installation. Simply select (or unselect) what you want to add (or remove), and the installation routine takes care of the rest of it. It works just like the Add/Remove Programs applet in Windows.

The other category are programs that are not included with your distribution. There are thousands of programs available on the Web, or perhaps you purchased a piece of software. (Don't laugh -it could happen!)

How to install one of these depends. In some cases, all that will be supplied is the source code and a set of instructions. In these situations, you'll have to compile and install the program yourself. It's definitely harder than having a user-friendly installer at your service, but it's not nearly as bad as some would lead you to believe. You're a programmer, right? You can type a couple of 'make' commands, right?

In other cases, the program already comes ready for installation. The files will be available as a 'tarball', which is sort of like a .CAB file in Windows, and the tarball may well be in a ".GZ" format - Linux's version of a ZIP file. Unzip the GZ file,

execute the tarball, and run the install program. You'll likely have to read the instructions the first few times, instead of just blindly looking for "install.exe", but it's not as much harder than Windows as just different.

Finally, you may run into a program that has its own installer built in - you just click on the file (or a link on a Website) and watch the thermometer bars zip on by until it's done.

## Top Ten Conceptual Hurdles with Linux

Linux is different from Windows. It's just like when you learned to program with a GUI, and had to develop a strategy for event driven programs, or when you began to program for the Web, and had to deal with stateless versus statefulness. There are some conceptual differences that you have to get used to. Here are my top ten:

1. Everything is a file in Linux. There's no differentiation between physical drives in Linux. With Windows, you're used to addresses drives with letters - A for a floppy, C for your first hard disk, D for your CD-ROM, and E, F and so on for additional local hard disks or remote shares. With Linux, all of these devices are simply directories under a single, coherent file system.

For example, suppose you have multiple physical hard disks in your computer. When you install Linux, it will choose which disks to install which components of the operating system and the applications. You don't have to know that the boot image is on one drive, that the application files are on another, and that your user data is on a third. You simply know that the boot files are in /boot, program files are in /bin or /lib, configuration files are in /etc, and your own data is in /home/<yourname>.

And other devices are located in the same file system. When you write to a floppy or read a CD, those devices have to be mounted to the file system first. Thankfully, those mechanics are usually taken care of for you, but it's still something to be aware of. All you need to know is that you find the contents of the floppy and CD-ROM under /mnt/floppy and /mnt/cdrom. And if you have a network share, that would typically be found under /mnt/q (or whatever you wanted to name your share - in this example, it's simply named 'q', just like a drive letter you could assign to a shared drive in Windows).

I won't get into the argument of whether this is better or worse - that's a value judgment. I find it to be cleaner to work with, but it took some getting used to.

2. Directory contents. As just noted, it takes a bit of getting used to what all of these strange directory names are. /etc, /root, /bin, /lib, /var, and so on. But this architecture, once you learn it, makes a ton of sense. Why did Microsoft think that intermixing system files, application programs, user profiles in the same directory structure was a good idea?

With Linux, these components are all separated. In fact, you can even set permissions on various directories differently, so that you can enhance security. Lock down your boot and program files directories, so that a bad guy can't try to take control over your machine, and only allow access to data directories by those folks whose data it is, for example.

3. Case sensitivity. Linux is a version of Unix. And Unix is case-sensitive. So MyDocument and mydocument are two different files. You've probably already been bitten by this if you've ever FTP'd files to a Unix Web server and found that references

between pages didn't work.

4. Root matters. "Root" is the name of the administrator under Linux. And Linux users tend to be much more disciplined about not logging on and doing things as administrator. Get into the habit of creating a user account and ALWAYS using it except when you absolutely, positively have to log on as root. Note that many operations that require root permissions will allow you to simply enter the root password to execute that operation, without having to log off as your regular user and log back in as root, and then reverse the process after you're done.

5. You don't have to compile everything. One of the urban myths is that with Linux, you have to compile everything before using it. I've been using Linux daily for a year, and I've only had to compile two programs - one that's in alpha stage, and the other was a small program I got off the Web. The large majority of the programs I use come ready to use.

6. Configuration is done through text files. I've already been through this, but it bears repeating. You can configure your entire machine through the editing of text files. It may take some time to figure out exactly what entries to use (do I use a "1" or a "2" to make the panel (the Linux GUI equivalent to the Windows task bar) automatically hide itself?) but that's all.

7. You don't have to reboot. This is one of the big changes in personal habits. As soon as something locks up, or even if things "just don't seem right", the natural Windows reflex is to reboot the machine. Linux comes from a legacy of stability and availability, and rebooting is as foreign a concept as sticking your hand into a socket to see if it's got power.

If something hangs up, look up the process ID and kill the process. Everything else will stay running because Linux applications just can't interfere with each other.

If you install a new application or reconfigure a service, stop and restart the service. You don't have to reboot when you update your Web server or install a new version of your browser like you do with Windows. The only time you must reboot is when you update the kernel itself - not the hundreds of ancillary services and programs that run on top of it.

8. You don't have to reinstall. Just like rebooting, reinstalling the operating system is a time-honored practice espoused by Microsoft as well as experienced developers. Things eventually just get "munged up" if you don't. And there are a host of incompatibilities between components that can force you to reinstall until you get the order of installation right. Linux, on the other hand, is built modularly, so that you can add and remove programs, and know that everything has been taken care of properly.

9. Security has to be addressed. It's 'common knowledge' that Linux more secure than Windows, right? You have to address security even though you're running Linux. Get your firewall installed, lock down your ports, get a virus scanner and an ad blocker, don't run as root, disable services you're not using, use hard to guess passwords. Practice safe computing practices regardless of the platform you're on.

10. You have a choice. Particularly in the United States, this may seem a funny concept to have to drill into people's heads, but remember that not everyone in the world has a choice of 31 dandruff shampoos and 20 kinds of bran cereals and a brew pub or a fast food joint on every corner.

With Linux, you have a choice of distributions. Of GUIs. Of application

software. Of utilities. The operating system does not rule your computer, it's there to support your use of it. When you install a new word processor or database program, you don't have a new browser jammed down your throat. When you update the kernel, you don't have a new version of your media player installed behind your back. You have a choice of what software you want to use, and for some, that takes some getting used to.

OK, one more for good measure.

11. Yes, Linux and open source software can be free, but it probably won't be. If you find value in the software you use, you very well may want to purchase the packaged version, and get some support included with it. Or you may choose to purchase service contracts on top of the software you've picked up. Or, scary as it may sound, you may choose to send some bucks the way of some company doing good stuff in the open source community, just to show your support. So while you can get by without spending any money, and for some constituencies (picture cash strapped college students, school districts, and non-profits), that's the best way to go. But don't be surprised if find yourself spending a few bucks in at least partial exchange for the value you're receiving.

## **How Fox and Linux Can Work Together**

There are a variety of scenarios for Fox and Linux working together. The first few scenarios involved Visual FoxPro running on Windows workstations and talking to Linux backends - this has been going on for a couple of years.

With the advent of the Wine project, you can run VFP on Linux, and talk to either Windows or Linux backends.

## **Installing Wine, Fox, MySQL**

### ***Installing Wine***

As of the writing of this paper, the shipping version of Visual FoxPro 8.0 was not available, and thus the testing of Wine with a production version was not possible. More details will be included in the updated version of this paper, available after the conference.

### ***Installing Fox***

As of the writing of this paper, the shipping version of Visual FoxPro 8.0 was not available, and thus the testing of Wine with a production version was not possible. More details will be included in the updated version of this paper, available after the conference.

### ***Installing MySQL***

MySQL comes with most popular distributions. If you didn't already install it, simply run the mechanism in your distribution for adding and removing programs. In Red Hat 8.0, this mechanism is found under System Settings | Packages.

## **Using Fox with Native Data**

As of this writing, Visual FoxPro will read and write to local DBCs and DBFs without any problems. There is currently a problem with record locking over the network, but it's one of the highest priority problems being worked on. When I started this paper, VFP couldn't do record locking even on local tables, and the estimate for a solution being found was several months - it was fixed in just a couple of weeks. I don't have many worries that record locking over the network will be resolved soon.

## **Using Fox with MySQL**

Connecting to a MySQL database on a Linux server from a Windows machine involves two steps. First, connecting the two machines in the first place, and then, second, using ODBC to connect to the database itself. More details will be included in the updated version of this paper, available after the conference.

## **Resources**

Running Visual FoxPro on Linux, by Paul McNett, FoxTalk, March 2003, June 2003  
[www.linuxtransfer.com](http://www.linuxtransfer.com)  
[www.openfox.org](http://www.openfox.org)  
[www.leaf.com](http://www.leaf.com)