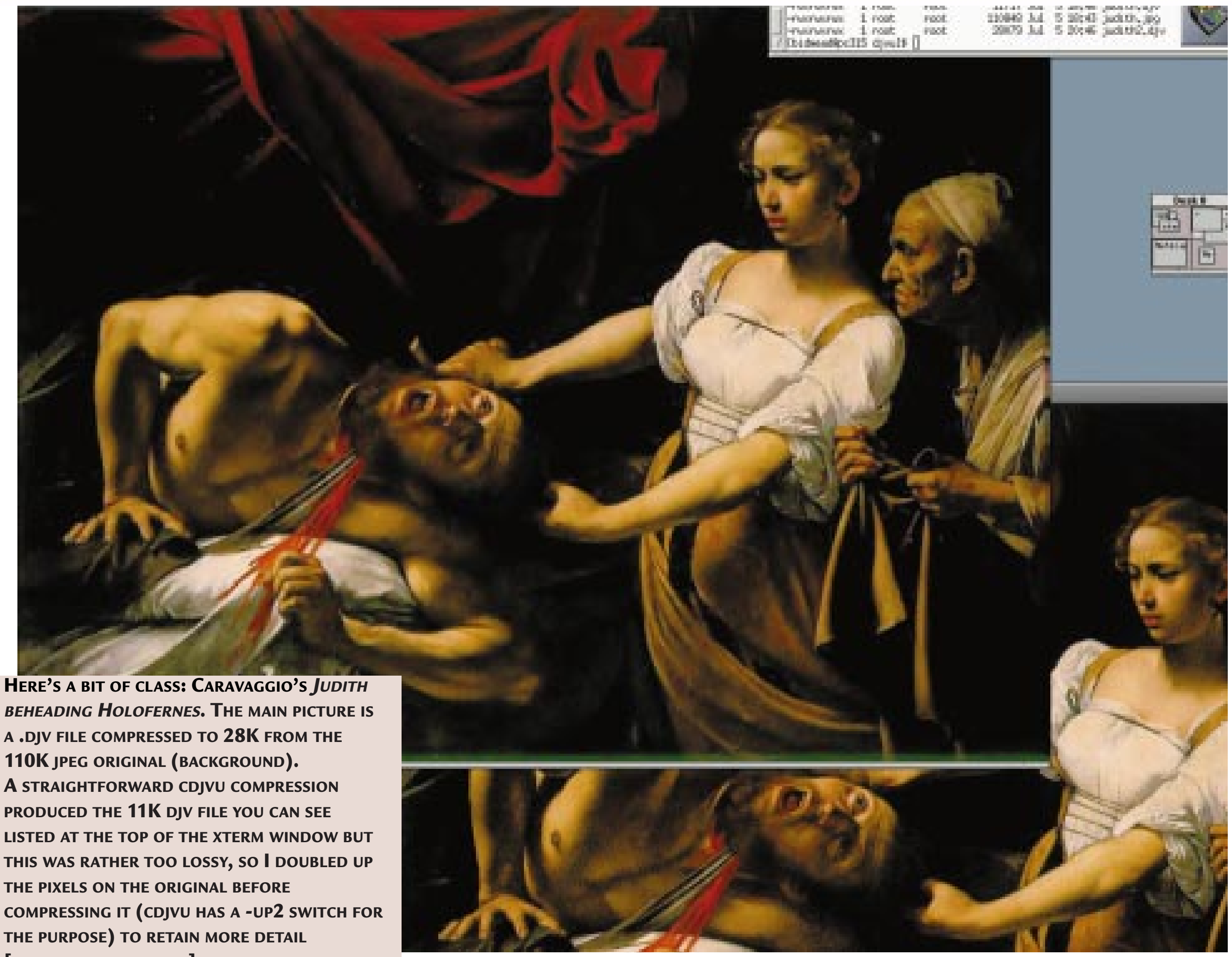


HERE'S A BIT OF CLASS: CARAVAGGIO'S *JUDITH BEHEADING HOLOFERNES*. THE MAIN PICTURE IS A .DJV FILE COMPRESSED TO 28K FROM THE 110K JPEG ORIGINAL (BACKGROUND). A STRAIGHTFORWARD CDJVU COMPRESSION PRODUCED THE 11K DJV FILE YOU CAN SEE LISTED AT THE TOP OF THE XTERM WINDOW BUT THIS WAS RATHER TOO LOSSY, SO I DOUBLED UP THE PIXELS ON THE ORIGINAL BEFORE COMPRESSING IT (CDJVU HAS A -UP2 SWITCH FOR THE PURPOSE) TO RETAIN MORE DETAIL [SEE MAIN TEXT, BELOW]

Picture this

DjVu is compression technology that aims to make high-res documents easily distributable over the net. Results, like the example above, can be impressive, says Chris Bidmead.

AT&T calls it the "next generation compression technology" and claims that it achieves compression ratios as high as 1000:1, which is five to ten times better than existing methods. DjVu (pronounced *déjà vu*) can squeeze a full-colour picture scanned at 300dpi down to less than 60Kb, instead of the typical uncompressed size of tens of megabytes. The idea is to make high-res colour and black-and-white documents easily distributable over the net.

You'll find a set of Linux utilities to handle DjVu at http://dejavu.research.att.com. AT&T recommends using a browser to display the compressed images, and includes a plug-in for Netscape. Alas, plug-ins are operating-system dependent and the one they supply only works with Windows. But you can display a compressed .djv image by using the ddjvu decompression utility and piping the result into Display, the utility that comes with ImageMagick. Chances are you already have this excellent graphics suite in your Linux distribution. If not, or if you're running a different platform, visit the home page at www.wizards.dupont.com/cristy/ImageMagick.html. You compress your original jpeg, gif or tiff (it also handles bmp, ppm and pgm) using the cdjvu utility. For example, to squeeze the Caravaggio (*above*), I used:

```
# cdjvu judith.jpg judith.djv
```

The ddjvu utility decompresses the image again, and you can feed it into Display like this:

```
#  ddjvu judith.djv | display
```

I tried putting that line into Netscape as a helper app but it seems not to like the pipe. If anyone knows how to make this work, do please drop me a line and I'll pass on the information.

● **Last month I may have** made getting X to work with the Matrox Millennium II sound more complex than it is. That's because I did it the hard way. The easy way is just to get a recent Linux distribution that uses XFree86 3.3.2 or later, because this has Millennium II support built into the standard XF86_SVGA server.

If, like mine, your Linux uses an earlier version of X you don't need to download and install the whole of XFree86 3.3.2. In theory it's only the server that needs changing. I got it from Suse at www.suse.de as xmatrox.tgz. It's just over 1Mb in size, so it's a manageable download.

Untar this with **tar xvzf X332SVGA.tgz** to extract the server. Now cd to the appropriate X directory. With my Caldera OpenLinux installation it's **/usr/X11R6/bin** but various Linux distributions still differ on this. An easy way to find it is to use the locate command to look for **XF86_SVGA**. Now rename your old X server (if it also happens to be called XF86_SVGA) and copy the new XF86_SVGA into this directory. Conventionally the server is evoked as X, so you'll need to establish a symlink. From inside the X bin directory you can do it with:

```
ln -s XF86_SVGA X
```

You can now check that the link is there with ls -1, which should show something like

```
X -> /usr/X11R6/bin/
XF86_SVGA
```

When X loads you don't normally run it directly. The usual way is to run a script called startx, which runs a binary called xinit, which in turn evokes X. When it loads, X consults a configuration file called **/etc/XF86Config**. You need to be sure this is set up with the right parameters to trigger the Millennium II features in the server. Unless you're an out-and-out geek you'll probably need the XF86Setup script to do this.

Earlier versions of XF86Setup don't know how to set the appropriate Millennium switches, so be sure to download the new version that goes with 3.3.2. The version on the Suse site is **xsuseconfig.tgz** and includes a database file called Cards. In my Caldera installation these two files go respectively into **/usr/X11R6/bin** and **/usr/X11R6/lib/X11**. You should check this first by using the locate command again to see the locations of your current versions of XF86Setup and Cards.

Now use the new **XF86Setup** to recreate **/etc/XF86Config** and you should be able to run startx as usual.

Ah, but there's another catch...

Nanny state

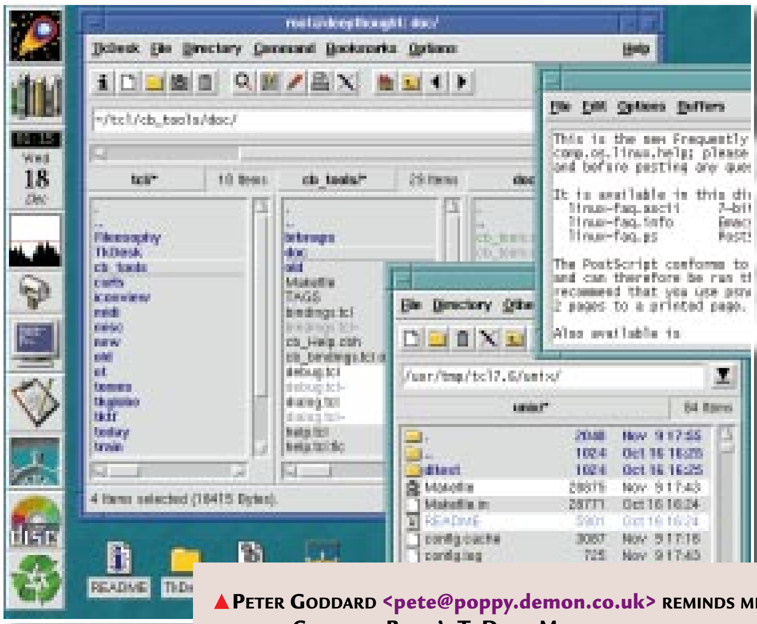
This will work fine as long as you're root, but if you're any other user you'll find X will refuse to load, with a message warning: "You should be using Xwrapper to start the server".

This is because XFree86 3.2.2 has suddenly become HAL in the film *2001* and is nannyishly warning you that you've hit a security issue. Xwrapper is a binary that XFree86 3.3.2 inserts into the startx->xinit->X sequence to get around this security hole. Unfortunately, even when I'd tracked down XWrapper I still couldn't run X as a user. The same warning kept coming up and I noticed it carried a rider: "We strongly advise against making the server SUID root!"

Because X has to do some hairy systems things, it needs to be running with root permissions. The point of Xwrapper is to ensure that these root permissions cannot be exploited by a mischievous user. As the only person on my network I made the judgment call that getting X to work was more important than guarding against security breaches, so I gratefully picked up the clue in the rider and made X the SUID root.

Insecurity guard

Setting SUID root on an executable is a bit complicated to explain but simple in practice. If root owns an executable file but allows me to run it (which is the usual case with Unix utilities), normally that file will run with permissions appropriate to my user status (rm, for example, won't allow me to delete root-


▲ Peter Goddard <pete@poppy.demon.co.uk> reminds me to mention Christian Bolik's TkDesk. Many readers ask me to recommend a GUI file manager — TkDesk is that and more. It runs across a number of Unix platforms, and borrows ideas from NeXT and the Apple Finder. "...and (shock horror) Windows 95, for some other (of course, minor and unimportant ;-)) inspirations," as Bolik puts it

(Illustration from http://people.mainz.netsurf.de/~bolik/tkdesk/)

[FIG 1]

Modifying permissions in X

```
# ls -l XF86_SVGA
-rwxÑxÑx 1 root root 3261116 Mar 7 11:46 XF86_SVGA
# chmod -v u+s XF86_SVGA
mode of XF86_SVGA changed to 4711  (rwsÑxÑx)
# ls -l XF86_SVGA
-rwsÑxÑx 1 root root 3261116 Mar 7 11:46 XF86_SVGA
```

[FIG 3]

Challenge Alex! Modifying mv with a simple script

```
#! /bin/sh
# mov- A simple shell script to move files or directories, with the added
# capability to move across file systems (which mv does not do).
#
# You'll have to just ignore the error messages as piping them to /dev/null
# seems not to work. You'd also better make sure the arguments are just
# simply "mov source dest" or bad things might happen like deleting something
# you don't want to delete.
if !(mv $@)
then
       cp $&
       rm -fr $1
fi
```

owned files). Setting the file SUID gives the executable full root permissions for as long as it is running.

This was the old, insecure way of running X, and I wanted it back! You can do it easily. Become root, switch to the directory the executable is in (or give its full path name in what follows) and then use chmod to modify the permissions. I like to use ls -l before and after to see the permissions. If you want to see them as they change you can use the —verbose flag for chmod, too [Fig 1].

Eye spy

Reader Neil Homer <neil.h@dial.pipex.com> writes that this column is to blame for pushing him off the straight and narrow (his job is administering Windows NT networks) in the direction of Linux. Sorry, Neil.

He calls Linux an eye-opener: "I bought RedHat 5.0 and installed it at home, and on a PC at work — much to my employer's annoyance. I was surprised at its ability to integrate into a multitude of environments. Strangely, developers who work in the same company were also going misty-eyed when I fired up our first Linux box."

Once he'd added ApplixWare he found that "If you can do it using Windows, then you can do it using Linux. I can't believe that I can do whatever I want with Linux and ApplixWare for about £140, that would otherwise cost me three or four times as much to do with Windows 95 and Office, never mind the rest of the server products that come with Linux."

Like many of you, Neil concedes that Linux probably has a long way to go before it's a direct challenge to Windows for the desktop. I've never seen it like this myself: if Windows does what you need, then fine; but if you want some control over your own machine, you should look elsewhere. Linux is a good place to start.

Package tour

Philip Sweeting <philip@jxs-software.demon.co.uk> writes: "Could you tell me what the extension 'rpm' signifies in Linux? How does one unpack it (I am assuming it is some sort of compressed file or something)?"

It stands for RedHat Package Manager. As the name implies, it was originally developed by RedHat (with financial support from Caldera). It's the standard package manager (method of installing and deinstalling apps, utilities and the OS basics) on these two platforms, and on others (like Suse), too. Go to www.rpm.org for more details.

Stamping ground

In the August issue I mentioned a tip from Andy Holyer <andyh@pavilion.co.uk> about using tar to move entire directories. Alex Holden <alex_holden@geocities.com> has emailed to point out that Andy and I have made a mistake in assuming that cp cannot be used to preserve file stamps. "The command line option you should use is -a or -archive which is analogous to -dpR. This copies a directory structure recursively and preserves as much as possible (including time stamps) in the process."

Alex is right, of course. But tar still has one advantage. Andy Holyer points out that adding the v flag to tar, as in:

```
# cd /var ; tar cf - spool |
(cd /somepartition ; tar xvf
 - )
```

prints out a log to the terminal of what's happening and, on a slow network "lets you know whether it's worth going for coffee."

Alex added a further thought which sparked a three-way discussion between us: "Mv won't move files between file systems because it doesn't actually move the files at all, but basically just modifies the directory entry.

"It sometimes annoys me when I forget that /usr/src and /home are separate partitions and I try to mv a file between them. Perhaps mv should be extended to test if the source and destination are on different file systems and, if so, perform a copy and delete action instead of a rename."

I told Alex I found the idea of modifying mv horrifying. "I hate the idea of complicating a long-standing and well understood utility," I said. "Particularly as it would be fairly simple to write a shell script that behaves as you suggest."

Andy agrees with me that the limitation of mv to a single file system can be a useful safety measure which warns that you may not be doing what you think you are doing.

Alex rose to the challenge with a simple script [Fig 3]. The brief was for a script that "behaves as you suggest", which would include error checking. Anyone care to improve on this?

PCW CONTACTS

Chris Bidmead *can be contacted via the PCW editorial office (address, p10) or email* **unix@pcw.co.uk**