



# Serial madness

**Chris Bidmead goes pear-shaped, overcoming serial problems while setting up comms drivers on Windows NT. And there's further Ntrigue, too.**

**M**y mate Marcus is getting quite fond of Windows NT: he's managing to do really useful things with it at work, setting it up Intranet-wise as the basis of an enterprise-wide information system and earning himself brownie points galore on top of his already enviable salary.

Alas, for me, things have gone so smoothly for him that he hasn't had to get embroiled in the entrails of NT, which is where I seem to be spending too much of my time lately. So he couldn't help me with the serial problem I had, and wasn't even able to calm me down when I got furious about the way NT handles this.

I was trying to set up the comms drivers on the Windows NT server running on the newly-installed NEC Powermate. I'm aiming to connect my network through it to Cix, CompuServe, Global Internet, Netcom, Pipeline and the seeming myriad other Internet service providers who've been offering me trial accounts recently.

Some, like Netcom, give it to you virtually on a plate, with installation from a CD-ROM that sets you up and connects you within minutes. The catch is that most of them assume you're going to be using Windows as your operating system, and at least one of them (Pipeline) won't work with anything else.

## Fiendish plan

Rather than hack away trying to get each one working with Unix, my fiendishly simple plan was to install them all on a Windows NT server and ultimately either use that as a router for the rest of my network, or migrate the more useful services to Unix once I was sure they worked well.

One small catch was that I couldn't get the serial port on the NEC Powermate to

respond. It turns out that there are better ways of going about the diagnosis than the one I chose, but as a Windows NT beginner the approach seemed logical to me.

So here's what I did. Open the Control Panel and click on Devices. This shows you a list of all the software and/or hardware hoojimaflips on your system, mixed in with a list of hoojimaflips that aren't on your system but which NT knows about.

The ones that aren't present are marked as Disabled and the others show whether they belong to the system, have been automatically started, manually started and so on. (Incidentally, in there I noticed the X11 display Network Interface that this version of NT adds to do its X magic. But more on that in a moment).

In there, is a device called Serial which is marked as automatically started. Select it, hit the Stop button, and a window pops up to tell you that the system is attempting to stop the serial device. After a moment the "Started" status disappears and you assume you've stopped the service.

OK, so now you hit the Start button. The system tells you it's attempting to start the service. About 20 seconds goes by and the word "Started" reappears in the Status column — so at least you know the underlying software service for the serial port is working and can proceed to the next bit of diagnosis. Yes?

No. I discovered that when the Devices module tells you the serial service has been started it may only be kidding. It turns out that my first port of call should have been the Event Viewer, a key utility you'll find grouped under Administrative Tools. There you get the real story. In this case it was a warning, which I quote verbatim: "While validating that COM1:

was really a serial port, the contents of the divisor latch register was identical to the interrupt enable and the receive (*sic*) registers.”

As it happens, after 15 years of messing about with RS232 devices, I understand what is going on. But frankly this jargon-ridden grammatical mess of a sentence, with its misspelling of "receive" really doesn't do much to reinforce your confidence in the system's view of itself.

This is followed by the peremptory: “The device is assumed not to be a serial port and will be deleted” — Oh, thanks. Thanks a bunch for not telling me about it and not even telling the poor old Devices module that was cheerfully reporting it had successfully started the service.

When the Unix people, with more than 25 years of their operating system behind them, say that Windows NT is “immature” I guess that this is the kind of thing they mean.

## Afterthoughts

Later... Calm down, Bidmead. Firstly, I now realise that the Devices module wasn't lying. It was accurately reporting that the Serial Service was being started and stopped. When it said it was running, it was. Distinguish between the "serial service", which is a software module, and the device itself (which is also a software module behaving as a virtual serial device a bit lower down the chain). The "device" wasn't working and was deleted. The service continued to run. Its decision not to report the disappearance of the device is, I suppose, a matter of taste.

That phrase of mine, about “15 years of messing about with RS232 devices” is somewhat double-edged in the light of what the problem turned out to be. There’s a classic joke about an old duffer who claims to have “15 years’ experience”, to which the retort is, “No. You’ve got one year’s experience. It’s just that it’s 15 years old.”

In my young day, a machine either had a serial port or it didn't. If I'd been paying attention I'd have remembered that about ten years ago manufacturers started introducing serial ports that could be switched on or off by software in the BIOS, to free up interrupts. Yup, the COM1 port was switched off (the plain English translation of "the divisor latch register was identical to the interrupt enable").

## NTrigue

My mention of the X Window System in conjunction with Windows NT may have puzzled you if you've just joined the



column. I'm running a product called NTrigue — essentially a version of Windows NT version 3.51 that translates the graphical user interface into X so that it can be squirted over the network. This allows you to pop up a Windows NT window (or several) on any machine on the network with an X display. NeXT natively uses Display PostScript instead of X, but a company called Intuitive Systems offers Cub'X, an X server application that integrates into NeXTStep as if to the manner born.

I told you last month that I would get NTrigue version 1.1 installed on the new NEC 100MHz Pentium Powermate in time for this column and for once I've managed to keep a promise, despite silly setbacks like the serial port problem. The installation is identical to a standard Windows NT install except for the process I mentioned last month, of having to run a validation program which provides you with a magic number derived from the machine and the date. You then fax that number to Insignia, which then faxes you back another number that unlocks the software — as long as you key it in before midnight on the same day.

Worrying, this. Picture a financial dealing room at around 11pm when the Tokyo market is getting under way and your 12 dealers all need their NTrigue sessions on their screens. Suddenly, for whatever reason, your NTrigue server crashes and has to be restored from back-up. Do you need to go through the validation routine with Insignia again?

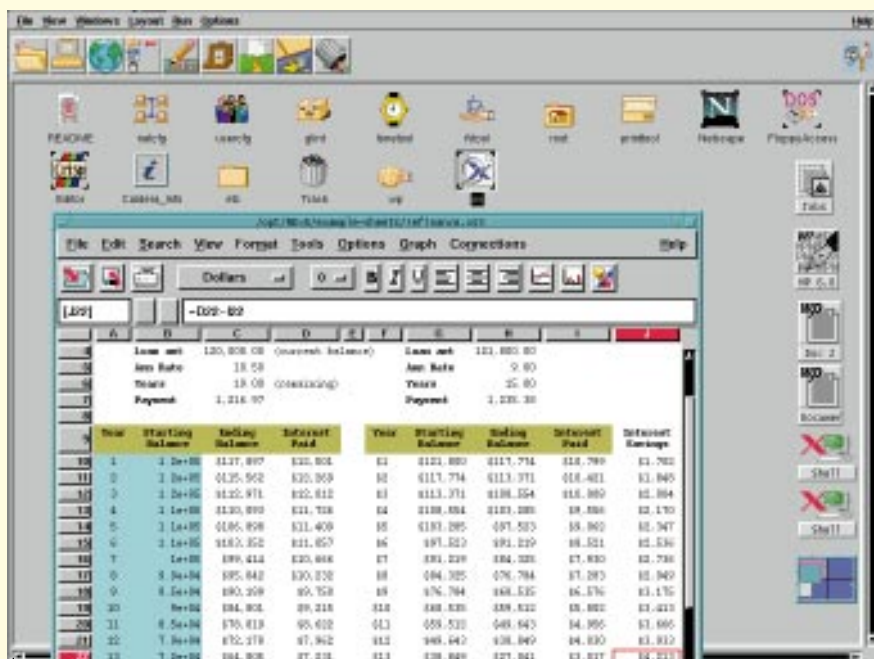
I checked with Insignia, which didn't have an immediate answer to this

*I've shown multiple system windows under X before but now that I can add Windows NT (thanks to NTrigue) the possibilities are becoming mind-boggling. This is the screen of my NeXT machine, with a window running the 32-bit version of Steve Palmer's AMEOL, with the Caldera desktop represented in a couple of other windows. The minimised Netscape icon you can see in the top lefthand corner also belongs to Caldera*

question. Worst case — what if the system restore has to be done at midnight from the original CD? In that case you'd certainly need to be able to get in touch with Insignia to get a new validation code. Does it run a 24-hour, 7-days a week service? Er... no, it doesn't.

With the main Windows NT engine installed on my NEC Powermate, the next step is to set it up to handle remote sessions. This is done with an extra NTrigue administration utility called WINCFG. You use it to create "WinStations" — individual connections to sessions on the NTrigue server.

NTrigue Winstations all use X11 as the transport mechanism, but can have different characteristics depending on whether you want to use encryption or allow sessions to time-out if left unused for a certain time. Winstations can be created in sets: essentially a collection of identical clone sessions with sequentially numbered names. You don't have to know these names when you log on, they're just automatically allocated.



*The latest addition to Caldera is the Internet Office Suite. It comprises WordPerfect 6.0 for UNIX (now under the aegis of Corel), NCD's Z-mail and a spreadsheet (seen here) called NEXS. Frankly, I don't think it's going to set the world on fire and without a database it's surely no great bargain at \$329. But the Caldera company sees it as the vanguard of a stream of UNIX commercial products that will be ported to Caldera over the coming months*

You log on from a remote X system by using the rsh (remote shell) utility which is a standard Unix feature also provided with Windows NT.

Instead of starting up a process in a shell on my own machine, if I enter the command

```
rsh <ProcessName> <YourMachineName>
```

I run the process on your machine. It's not quite that simple, though, as there will usually be some security hurdles to be cleared, but you get the general picture (man rsh will tell you the full story if you're running Linux or something similar). NTrigue uses this command to establish a Winstation connection, and on my system it goes something like this:

```
rsh powermate nextmachine:0 640x480 4
```

where "powermate" is the name of my NTrigue server; nextmachine:0 is the X display I want the session to appear on; and the figures at the end (you've guessed it), define the size in pixels of the display and the colour depth. The size is arbitrary. I've got the habit of creating windows of different sizes depending on the Windows NT applications I'll be running in them.

### Not so simple copy and paste

So that's how I get to run Windows NT on my NeXTStep desktop. Necessarily there are some distinct oddities about this, because the cultures of the two operating systems are completely different. But I've got used to this sort of thing from running OS/2 and its built-in WinOS/2 environment. The NTrigue setup costs you an extra machine but in return gives you a full 32-bit version of Windows. And X lets me add Caldera (or whatever) to the mix too.

This set-up saves me walking across the room between the Windows NT machine and the NeXT machine but it's hardly the integrated environment of one's dreams. I'm hoping OpenStep for Windows NT will bring me closer to that in the fullness of time. Meanwhile I have to make the best of X's limited talents at text-only copying and pasting between screens. For graphics, you have to use screen-grabbing.

At first, I couldn't get copy and paste to work. For example, I access the online news service NewsBytes through the Cix conferencing system and connect to Cix using Steve Palmer's AMEOL off-line reader. If there's an item in NewsBytes that I want to save in my NeXT machine's Digital Librarian, I copy it in the NTrigue window and paste it into the NeXT editor. Well, that's the theory. Except that nothing, or something completely different, would appear in the NeXT editor.

The copy and paste features in X can be very baffling if all you're used to is the Windows way of doing things. Essentially, instead of having just one clipboard, there are several. In IBM's implementation of X

for OS/2, called PMX, the documentation describes the cut buffer as a stack of eight separate buffers: "By convention, data first goes into CUT\_BUFFER0. As each new piece of data is added to the cut buffer, the existing data is moved to CUT\_BUFFER1, then CUT\_BUFFER2, and so on."

This process is reversed when text is transferred back from the cut buffer stack. The catch is that the stack handling isn't managed centrally by the X server. Each X client is supposed to take care of rotating the data in an orderly fashion, but in practice they tend to do this in different ways.

Just to add some spice to the story, X11 release 3 introduced yet another cut buffer: a special one called PRIMARY. The principle of PRIMARY is that if you've dragged the mouse over some text it's probably because you want to do something with it. So even if you don't bother to do a copy, any text you select automatically turns up in PRIMARY.

In order for NTrigue to be able export the contents of the Windows NT pasteboard, the installation routine puts a special utility called NTutil into the Windows Startup folder. NTutil automatically copies the contents of the Windows clipboard into PRIMARY, which is hardly logical as it can only arrive there after an explicit copy. On the other hand, the CubX people have used CUT\_BUFFER0 as the transfer buffer between X and NeXTStep. This is a sensible choice because NeXTStep uses copy and paste extensively and, like Windows, doesn't use the PRIMARY style of short cut.

So there's the problem: text copied from NTrigue moves into the X window's PRIMARY buffer, but text copied into NeXTStep is expected to arrive from CUT\_BUFFER0. The train turns up on the wrong platform.

My roundabout solution was to start up an X editor under CubX (which comes with a full suite of the standard X mini-clients compiled for NeXT). I'd copy from a Windows app running under NTrigue, dump the text into the X editor by clicking the right mouse button, then select the text, copy it, (which gets it into CUT\_BUFFER0) and thus be able to get it into NeXTStep. There's got to be a better way, I thought, having done this a couple of times. Happily, there is, but more about this next month, and the new developments on the Caldera front. ■

### PCW Contacts

**Chris Bidmead** is a consultant and commentator on advanced technology. He can be contacted on [bidmead@cix.compulink.co.uk](mailto:bidmead@cix.compulink.co.uk)