



# Memories are made of this

Andrew Ward advises **Athlon users** to install SP6, and checks out physical memory size.

**A**lthough new features are no longer included in new service packs from Microsoft, there is one small exception. Prior to Service Pack 6, the Windows NT kernel did not turn on the write-combining feature on the AMD Athlon processor. When the system starts up, Windows NT checks for the processor type, and Microsoft says if it detects a Pentium II or III chip (even though the feature is also present on the Pentium Pro), then it enables write-combining. Now with SP6, this check will also detect an AMD Athlon chip and correctly enable write-combining.

This speeds up video processing. Usually, many writes to video memory are individual pixels, which constitute a single 8-, 16- or 32-bit value rather than a full cache line of 32 bytes. However, the hardware architecture is optimised for burst transfers of full cache lines: writing data 16 bits at a time is much slower. Write-combining uses internal processor buffers to combine smaller or partial writes into full cache lines and hence into burstable writes.

If you have an AMD Athlon, upgrade to SP6 if you haven't already. The performance improvement could be as much as 15 per cent for graphics operations.

## ■ Memories

For those of us who spent years battling with memory managers like QEMM, DOS memory might still be of some passing interest. However, I'm sure Michael Rickard is correct when he writes in to suggest that what Paul McCormack (January issue) really wants to know about remote machines is how much

physical memory they have, and not how much conventional and expanded memory is available to a DOS program. Unfortunately, my suggestion of using the MEM command therefore isn't going

*As far as I know there is no command-line utility to return physical memory size*

to be of any help, and as far as I know there is no command-line utility to return the physical memory size.

Michael suggests a couple of tricks, however. The first, assuming you have command-line access to the remote machine via rcmd or some other mechanism, is to use the facility of winmsd to dump its data to an ASCII file. Simply type the following command:

```
winmsd /s /f
```

and it will write the system data to a file named %computename%.txt in the

current directory. This does work, although you will then either have

to read the file manually or write a simple script to parse the result.

Another technique is to use the regdmp command, although once again you'll need the resource kit for this (it's

also included in Microsoft software development kits for NT). Unfortunately, not only does this produce rather a lot of output, you'd also have to manually add up the three memory sizes that are displayed under the physical memory key. Michael provides a script to make life easier for you, shown in Fig 1. To use this script, type:

```
rmemsz \\remotepc
```

If you use it without any parameter, it will return the physical size for the local machine. Using regdmp assumes you have administrator rights on the remote machine. If you don't, it won't allow you access to the remote registry.

For those of you who aren't command-line wizards, a few words of explanation might be in order. Type **FOR /?**

to check out the FOR command – if all you're used to doing with it is typing things like:

```
for %a in (*.cmd) do find "regdmp" %a
```

FIG 1

**rmemsz.cmd** – a command file to return a remote system's physical memory size

```
@echo off
rem get the physical memory size of a remote PC running NT4
rem requires access to the registry
rem uses regdmp (from resource kit or sdk or ddk) as this dumps
rem the REG_RESOURCE_LIST that is returned in a readable form
rem the script assumes that all LENGTH items are RAM
rem which is usually true
setlocal
set memkey="HKEY_LOCAL_MACHINE\hardware\resource\map\system
resources\physical memory"
set mc=-m
set pc=%1
if not defined pc set mc=
set memsz=
for /f "eol=# tokens=5" %a in ('regdmp %mc% %pc% %memkey%
^| find "LENGTH"') do (set /a memsz = memsz + %a / 1024)
rem if find didn't succeed bail out (usually regdmp failed,
and will have
rem written an error message to stderr
if errorlevel 1 goto :eof
if not defined mc set pc=local machine
echo %pc% reports %memsz% KB RAM
endlocal
```

(Key: ✓ code string continues)

then some of the constructions will be unfamiliar to you. First, eol=# is actually a dummy value – eol is required by the FOR syntax, although there aren't any lines within the output we wish to ignore.

Next, tokens=5 takes the fifth token from the line being parsed and places it into the variable %a. To distinguish between tokens on the line, white space is used as the delimiter because we haven't specified anything different. The fifth token is therefore the hexadecimal value that appears after LENGTH.

The /A switch to the SET command tells it to evaluate a numeric expression – necessary in this case because we want it to add the three LENGTH values into the memsz variable. A by-product of using the /A switch is that it is quite happy to accept values in hexadecimal notation, but the variable's value is ultimately reported to us as a rather more user-friendly decimal value.

A minor note is that variable names need to be preceded by two percentage signs when used in a batch file, whereas a single percentage sign is sufficient when you use FOR from the command line as in the example above.

FOR is designed to be used to process a set of files, but by enclosing a command – or sequence of commands – in single quotes, FOR will parse the output of those commands instead. The command FIND "LENGTH" is used to ensure that we only parse lines that contain the word LENGTH. In fact, we get the same result if we omit this, since the other lines either don't contain a numeric value in the fifth token, or it's too small to affect the result.

Now, I've left as an exercise for you to work out a similar script to parse the output of winmsd – it's really only the

FOR command that's of interest. You can also work out why the script shown in Fig 1 only works as a CMD file and not as a BAT file.

## ■ Sounds familiar

Stuart Taylor upgraded his system from Windows 98 to Windows NT4, but then discovered that once his Creative Labs Sound Blaster Live! drivers were installed the system would no longer boot.

Stuart found the Creative driver he was using didn't work with Service Pack 4, and so he downloaded a new driver from the Creative website. I'd just like to comment that this sort of occurrence isn't particularly unusual – so much so, that in general if I buy a piece of hardware I throw out the CD that comes with it and go straight to the web to look for new drivers (although not necessarily in that order). It's also a good idea to periodically check out the hardware vendors' websites for updated drivers for the hardware you already have.

## ■ Printing woes

Unfortunately, the downloading



**SCREENSHOT 2** USE THE WINDOWS NT XJ6C DRIVERS FOR THE XEROX DocuPrint C6

technique doesn't always work. A popular Internet shopping site for computer bits and pieces recently offered a colour printer for less than £50. Linda Davies was among those who took advantage of the offer, but was dismayed to discover the Xerox DocuPrint C6 doesn't come with Windows NT drivers, and there aren't any on the Xerox website either.

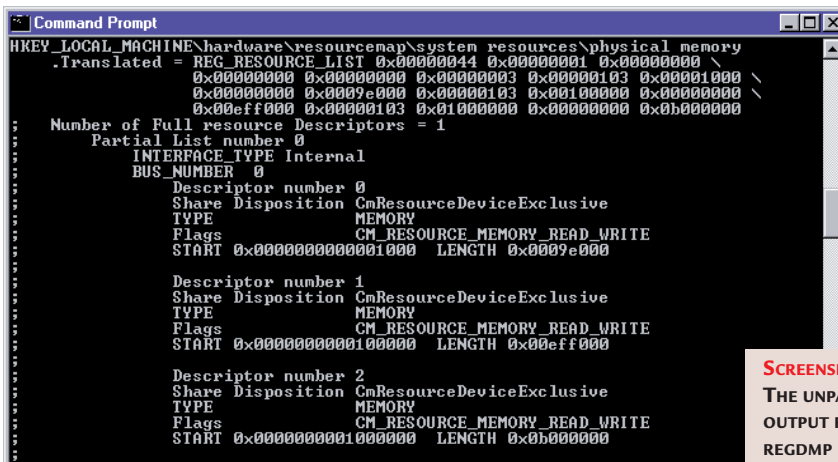
As usual, the technique of trying drivers for printers with similar-sounding names proved to be a useful workaround. So if you have one of these printers, download the driver for the DocuPrint XJ6C and try that instead.

## ■ Default system locale

A very common problem, and one raised this time by Richard Carey, is that of the default system locale for new users.

I'm sure we've all tried going to the Regional Settings control panel, ticking the box that says Set As System Default Locale, and then wondering why all new users still have their regional settings configured for English (United States).

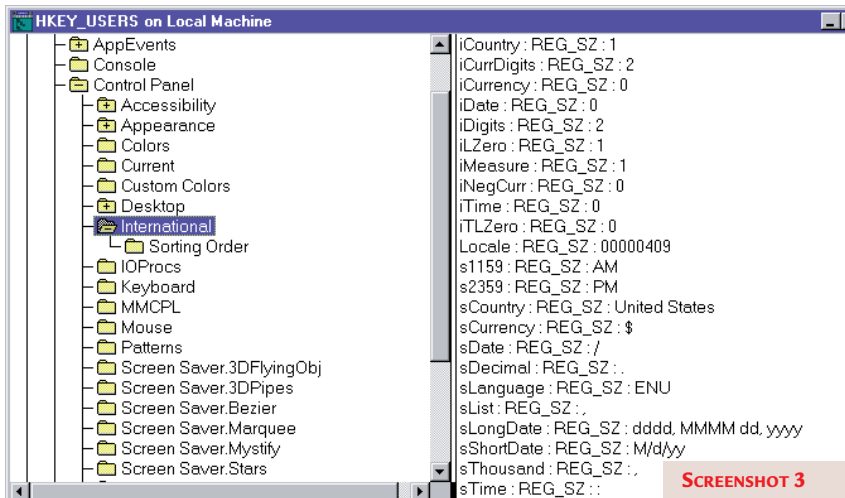
There is a way to solve this problem, and it's a question of modifying the NTUSER.DAT registry hive for the default user. This default hive is copied over whenever a new user logs on for the first time, and it will be in this folder: %systemroot%\Profiles\Default User



**SCREENSHOT 1** THE UNPARSED OUTPUT FROM REGDMP



# hands on windows nt



First, take a backup copy of NTUSER.DAT file by copying it to NTUSER.DAT.BAK. Then start up REGEDT32 and select the HKEY\_USERS window. Select the Registry/Load Hive menu entry and then navigate to the default user NTUSER.DAT file to load it in. When asked to give it a key name, choose DEFAULTUSER. Then select the DEFAULTUSER key and navigate to the values you need to change. In this case, go to: DEFAULTUSER\Control Panel\International. There you'll find lots of entries, like iCurrDigits, and you'll probably have no idea what to set them to. Therefore, it's a good idea before you get to this point to first navigate to the International key of a user you've already set up with the appropriate regional settings for your country.

Once you've tracked them down, from the Registry menu select Save Key and choose an appropriate location and filename (UK, FR, BE and DE are good examples of filenames to use). Everything in the International key will be saved to disk.

Now, go back to the DEFAULTUSER International key, and from the Registry menu choose the Restore option and select the appropriate file. The country settings will be read in from that file. Of course, you could make a note of the desired settings and make the changes manually, but that's rather long-winded.

To save the changes you've made, go and select the DEFAULTUSER key again

and now the Unload Hive option on the Registry menu will reappear. Select it, and the NTUSER.DAT file will be written back out to its original location.

What we've done so far will merely change the default locale for all new users created on this particular Windows NT server (or workstation). If we want to extend the changes throughout a domain, we need to set up a default user profile in the Netlogon share of the primary domain controller. Usually, this will be found at: %systemroot%\System32\Repl\Export\Scripts

The easiest way to do this is to make a new user called Fred and configure

his profile to exactly how you want the default profile of all new users to be.

Then, using the System control panel, copy Fred's profile to the Netlogon share. Note that no directory name will be created – you'll have to append \Default User to the end of the path to Netlogon. Also, before you click OK, click Change to set Permitted To Use to Everyone.

You've now set up a default profile on the primary domain controller, but it should also be copied to other domain controllers that are involved in authenticating user logons.

#### ■ Slow down you move too fast

Mark Baugh has successfully upgraded a number of PCs, but unfortunately they're

now too fast!

Apparently, an article on Microsoft Technet says applications written in FoxPro V2.5a won't run on anything faster than 333MHz, and Mark has gone and replaced aging 33MHz 486 PCs running Windows 3.11 with 400MHz Dells running Windows NT4. Now, the planned maintenance system written in FoxPro won't run.

This is an age-old problem. When people wrote software – in particular games – for the original IBM PC, they frequently built in some sort of dependency on the PC speed. The utility I'm going to recommend – Mo'Slo – was originally created to allow people to run these games on the new, faster 386 processors. Once software authors realised that processor speeds were on the increase, most ceased to include any speed-dependent code, but obviously that message didn't get through to the FoxPro developers!

Mo'Slo is available from [www.hpaa.com/moslo](http://www.hpaa.com/moslo). It works within a command session and will slow down any .com or .exe executable to the specified percentage of the current CPU speed. Hence the following command line will run GAME.EXE at 10 per cent of system power:

```
moslo /10 game.exe
```

There are three versions of Mo'Slo – a free trial version, the Deluxe version and a pukka business version that allows you to specify an absolute speed that will work across a variety of systems, rather than having to determine the appropriate percentage figure for different CPU speeds. Also, the default speed of the free version is a 4.77MHz IBM PC, whereas that of Mo'Slo 4Biz is a 166MHz Pentium which is adequate for most purposes, and probably just right for FoxPro. The 4Biz version is also able to slow native Windows applications in Windows 95 and 98 and has an optional hotkey feature to allow you to turn the slow-down on and off at runtime.

## PCW CONTACTS

Andrew Ward welcomes your comments on the Windows NT column. Contact him via the PCW editorial office or email [nt@pcw.co.uk](mailto:nt@pcw.co.uk)

Mo'Slo Deluxe costs £12.50 and Mo'Slo 4Biz costs £14. Both can be purchased online from:

<https://secure.reg.net/product.asp?ID=1041>