The mousetrap

Yes, it's a good old-fashioned whodunnit: when your mouse trips up while treading the, er, mat, you can help it get its act together. Panicos Georghiades and Gabriel Jacobs direct.

With standardisation on Microsoft-compatible mice, mouse problems have lessened but haven't disappeared. Mice can still fail to function at all, be too fast, too slow, move in certain directions and not others, and work in some programs but not others.

Mouse matters come in threes. There are three types of mouse: Microsoft, Microsoft-compatible, and Microsoft non-compatible; three different ways of connecting them: via the bus, a PS/2-style socket, and a serial port; and they can be used in three types of application:

1. DOS applications, in which case they usually require a driver loaded in the config.sys or autoexec.bat files.
2. Windows applications, in which case the driver is provided by Windows, or a Windows driver is provided by the mouse manufacturer.
3. DOS applications running within Windows.

And, yes, there are three different causes of mouse problems:

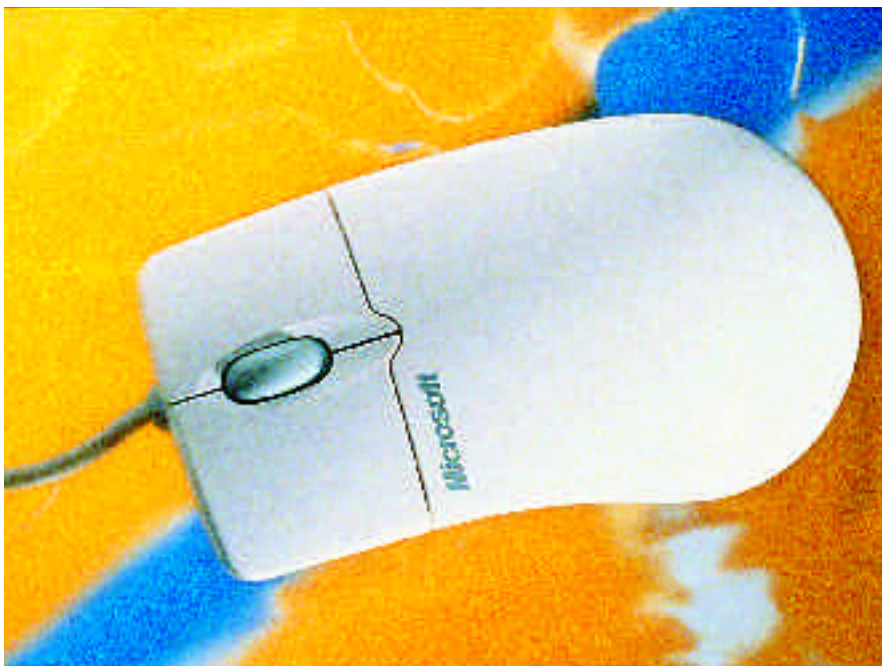
1. Hardware: the mouse isn't plugged in properly, it's a bad or unclean mouse, there's a bad mouse socket, a bad cable, a slippery or uneven mouse mat, or the mouse is connected to the wrong port or with a wrong or bad adaptor.
2. Bad drivers: an older or incompatible version is being used.
3. Conflicting software: your mouse driver may not agree with Windows, or a particular program, or with some other program running at the same time like a TSR program, anti-virus software, or screensaver.

A classic situation is using too many drivers. Manufacturers provide drivers for Windows and DOS (and sometimes for particular DOS programs) and nowadays for Windows 95. Don't install them all — Windows may not run properly if DOS mouse drivers are also there.

So, if you have a mouse behaving badly, first check for hardware causes — plug it into another computer, or plug another mouse into your computer. Next, establish that the mouse works in DOS and in Windows separately. If the mouse doesn't work in a DOS application under DOS, it will not work in that application if you run it under Windows. Install the DOS drivers needed in the autoexec.bat and/or config.sys files, as described in the mouse's documentation, and ensure that all works fine under DOS.

If you have a Microsoft mouse, use version 8.2 of the driver which comes with Windows 3.1 and, if necessary, expand mouse.sy_ and mouse.co_ (they're on the

p264 ➢



When problems strike, are you a man or a mouse? Take valuable advice on failsafe fix-its

Adjusting mouse settings

- Lowering the Sensitivity value in the Mouse section of Control Panel makes the mouse movements less jumpy.
- If the mouse is jumpy in Program Manager group windows, lower the granularity settings in the Desktop icon of Control Panel.
- If you're using the DOS-based mouse driver mouse.com or mouse.sys version 7.04 or later, add the /Y switch to the end of the mouse command line

```
(c:\windows\mouse.com /y)
```

- Note that erratic mouse movements may be specific to the application, video card, machine BIOS, keyboard BIOS, or machine type you're using.

Windows disks) to your hard disk as mouse.sys and mouse.com by using the EXPAND command at the DOS prompt. You can test whether all works well using a DOS program which supports mouse movements such as Edit. Then disable the DOS drivers by placing the word REM in front of the lines referring to them in the config.sys and autoexec.bat files, and check the mouse works under Windows.

The Windows mouse drivers are set using Windows Setup, and they appear in the system.ini file in the [Boot] section. Normally there should be a line like

```
mouse.drv=mouse.drv
```

To check you have installed the right Windows driver, exit Windows, change directory at the DOS prompt to c:\windows, and type SETUP. If you get a message saying no mouse has been detected, select the Microsoft or IBM PS/2 option.

Note that some so-called Microsoft-compatible mice are more compatible than others, and you might have to use some trial and error. In particular, try the drivers which come with the mouse installation disk rather than the Microsoft drivers. If all the above fails, try the following suggestions.

- Search the drive for multiple mouse.drv files. If you find any, rename them to something else, except for the one in the Windows System sub-directory.
- Test the mouse on a different port.
- Check that there's only one mouse.ini file, and that the line

```
MouseType =
```

in the [Mouse] section of the file points to the correct port.

- Try running Windows in standard mode. If all works well, try loading Windows by typing win /d:x. If all is still okay, add the following line to the [386Enh] section of system.ini:

```
EmmExclude=A000-EFFF
```

- Finally, if you're using a mouse that came with its own drivers, try to borrow a mouse that uses the driver supplied with Windows. If that works, contact the manufacturer of your own mouse.

Out of Africa

"I am doing voluntary work in Nigeria, and have been working on a stock-control and tracking program using DOS 6.22 and QBasic (the v4.5 compiler), but the compiled version of the program gives a totally incorrect output at the printer — the numbers are all wrong. Inspecting the code doesn't reveal much, as the only difference lies in an LPRINT statement (to printer) and a PRINT statement (to screen). There is no problem with interpreted printed reports or displayed reports. I've tried two different printers — no difference. I enclose some output samples. My big fear is a virus, as they are rampant here."
Leigh Bowden, Nigeria

The Microsoft Knowledge Base doesn't list your problem, but there are a number of possible causes.

We doubt a virus is the culprit — virus programmers tend not to target something as old as QBasic! We assume that you're not trying to print while running the program in a DOS session within Windows, as this doesn't work with all DOS programs.

The output samples you sent us show that the text prints fine; only the numbers are wrong. This obviously implies that the problem lies in calculations or statements to do with numeric data only — perhaps a different set of calculations is being used to output to the screen than to the printer. Programmers sometimes attach calculation statements to PRINT statements, and since QBasic needs two different statements, one for the screen and one for the printer, the set for the printer may contain errors.

Alternatively, it may be that certain formatting commands used for the printed output don't work properly when compiled. In any case, check you're using the right version of the compiler for the version of QBasic you have, and that you're setting the right options for handling numbers for the compilation. Also, have you checked the compiler documentation for supported statements and commands? Some statements are supported by the interpreter but not by the compiler.

The write font

A frustrated Bill Reid (mailto:reidw@nacn.dnet.co.uk) mailed us about being unable to change the default font in Windows Write. Each time you start Write, the default font is Arial: he wants it to be Times New Roman.

Trying to answer this query has prompted us to write about a method of creating templates for Windows Write, to use for letters, memos, and faxes, a feature available on all mainstream word processors but missing from Windows Write.

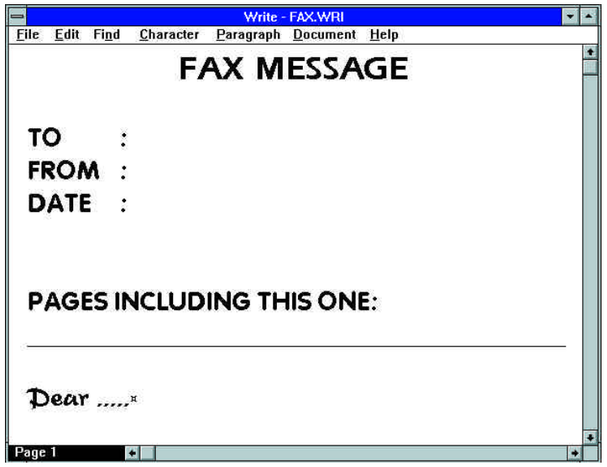
You can create your own designs and save them as standard Write documents. Give them names like letter.wri, memo.wri and fax.wri. To prevent overwriting the templates, you have to Save As, not Save, and to avoid using Save by mistake, make your template files Read Only by changing their Attributes in the Properties option of the File menu in File Manager.

To use the templates you open Write, then use File Open to select a template, or you can assign icons to each template so you can open them with a double click. To do this, start Write and the template in one go by editing Write's Properties in the File menu of Program Manager. In the command line of the Properties dialog box add the filename of your template, so the command line reads, for example, write.exe letter.wri.

Do this for all your templates by clicking and dragging the Write icon while pressing the CTRL key to make copies of it, then editing each one's command line to include a different template filename. Change the default Write icon to another one by using icons embedded in Progman.exe.

To change the default font from Arial to something else, create an "empty" template. The only problem is that the template can't be completely empty, so you need to type a space, say, then select that space with the mouse, and from the Fonts menu change the font to the one you want.

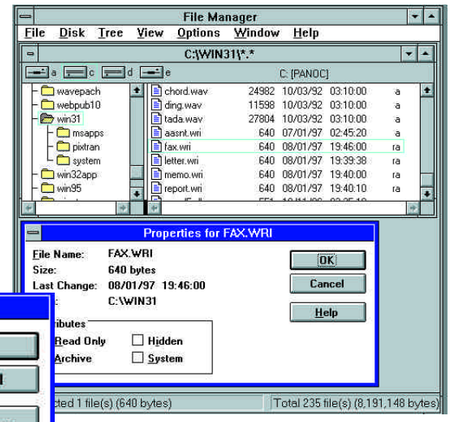
A step-by-step guide to creating Write templates

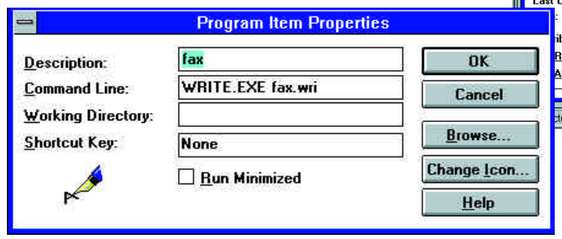


Above Create your templates in Write and save them as any ordinary Write documents

Right From File Manager make the template files Read only

Below Create multiple copies of the Write icon and edit each one to include the template file name in the command line





Right You will end up with multiple Write icons, each one starting a different template

Below To customise your templates even more, choose a different icon for each one using the icons embedded in Progman.exe, or any other icons you may have

PCW Contacts

If you have any queries or Win3.1-related topics to discuss, contact Panicos Georghiades and Gabriel Jacobs at Win3@pcw.vnu.co.uk.