

Inside Microsoft Windows: NETWORKING EDITION

Tips & techniques for Microsoft Windows

Avoiding major catastrophe by using the WINDIR variable

By **Chris Morton**

You should never run certain utilities and applications when Windows is active. Some disk maintenance programs and virus detection tools fall into this category. Additionally, many automated batch files (*.BAT) may operate fine over a network when systems are idling at a DOS prompt, but they'll crash and burn if they're engaged when Windows is active on a PC.

A safety net—with holes!

Some unsuspecting users may think it's okay to run one of the programs mentioned above at what they think is a standard DOS prompt, only to discover later—when their files are being reconstructed—that they just shelled out to a full-screen DOS session by double-clicking Program Manager's MS-DOS Prompt icon. Unbeknownst to them, Windows was still running! To counter this problem, savvy administrators know to add a SET WINDIR= statement to the AUTOEXEC.BAT file of each system so it reads something like

```
SET WINDIR=*** Warning! *** Windows is still active.%_sp%_sn%$g
```

When shelling out to DOS from Program Manager (after rebooting the system), the user now sees a prompt similar to the one in **Figure A** that advises

```
*** Warning! *** Windows is still active.
```

Using a custom shell prompt is certainly better than relying on the standard DOS prompt, but it can't prevent a user from issuing a destructive command. Fortunately, there's a way to protect users from themselves.

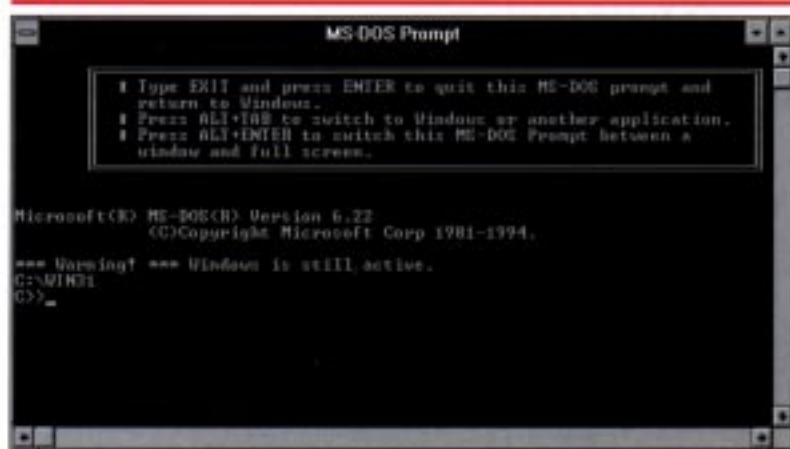
Testing for WINDIR

Under Windows 3.0, there really was no effective way to find out if Windows was active. To eliminate this concern, Microsoft heeded system

administrators' requests when developing Windows 3.1 by including a unique system variable—WINDIR. It exists in the DOS environment only when Windows is active and is deleted when Windows is shut down.

The ultimate solution, then, is to create a "wrap-around" batch file for each command that poses a threat to the system. The batch file tests for the existence of

Figure A



You can make your DOS prompt a warning to inexperienced users.

IN THIS ISSUE

- Avoiding major catastrophe by using the WINDIR variable 1
- Emulating Windows 95's taskbar in Windows 3.x 3
- Plug-In provides a powerful Windows enhancement 8
- INI 108: SYSTEM.INI's [NonWindowsApp] section prepares DOS applications to run windowed 10
- Networking tips 13
 - Troubleshooting cable problems on your network*
 - Sharing violations*
 - Outdated drivers cause lockups*
 - Problems with Microsoft At Work Fax*
 - Problems with changing your password in Mail*

WINDIR; if it detects the variable, the batch file exits gracefully to prevent system hangs, total disk corruption, and other calamities of varying degrees.

The wrap-around batch file, shown in Listing A, relies on detection mechanisms included in

newer versions of the FIND command. You should use this technique only with DOS 5.0 or later. You may use any text editor, including the MS-DOS Editor or Windows' Notepad, to create batch files that incorporate the wrap-around code. If you use your word processing program, be sure to choose Save As... from the File menu, saving the batch file in text (TXT) mode to avoid the formatting codes your word processor inserts automatically.

LISTING A

```
@ECHO OFF
REM Note that EL is used below as an abbreviation for ERRORLEVEL.
SET > DETECT.TXT
FIND /I "windir" DETECT.TXT > NUL
IF ERRORLEVEL 2 GOTO error
IF ERRORLEVEL 1 GOTO continue
REM If the command sequence reaches this point, the FIND test
REM has detected that Windows is running and has set
REM ERRORLEVEL=0.
CLS
ECHO This batch command cannot be used when Windows is running.
ECHO Exit Windows and run this batch file from a DOS
ECHO prompt.
GOTO exit
:error
REM An error has occurred during the FIND test.
CLS
ECHO Alert the system administrator that an error has occurred
ECHO on your system when this batch file was run.
GOTO exit
:continue
REM Windows is not running.
CLS
REM Substitute the actual command series you want to run on
REM this PC here. Be sure to remove these REM statements.
:exit
DEL DETECT.TXT
```

How it works

As the SET command starts, it reports the contents of the current DOS environment. Instead of displaying the information onscreen, however, the data is redirected (>) to a file named DETECT.TXT. The FIND command then inspects this file for the existence of the WINDIR environment variable. The /I switch instructs the FIND command to ignore case sensitivity.

If Windows is running when the batch file is engaged, the FIND command will locate the WINDIR environment variable in DETECT.TXT. FIND sets the ERRORLEVEL code to 0, so the command sequence ignores the two IF ERRORLEVEL traps. (To learn more about ERRORLEVELs, run Help from DOS and click the If command.) The current display then clears, and the ECHO statements notify the user that he or she shouldn't engage the batch file while Windows is running.

Control then passes to the end of the file, where DETECT.TXT is deleted.

If an uncommon error prevents FIND from functioning correctly, the command sets the ERRORLEVEL flag to 2. This value meets the condition set by the first trap, IF ERRORLEVEL 2, and control passes to the :error section. After the batch file clears the screen, ECHO statements display a suitable error message to the user and control passes to the end of the file.

If the system is at the DOS level when a user engages the batch file, the file gets the green light. FIND sets the ERRORLEVEL flag to 1, passing control to the :continue section. It's here that you should insert the actual command sequence you want carried out. Since this condition means Windows isn't running, you may launch almost any program without concern.

Added protection

If you want the batch file to run a program named ANTIVIR.EXE, for example, you might save the completed batch file as ANTIVIR.BAT. Place the batch file in a directory, such as C:\BAT, that you have specified in the DOS PATH statement. Your PATH statement, found in the AUTOEXEC.BAT file of the system's root directory (C:\), might now look something like

```
PATH=C:\WINDOWS;C:\DOS;C:\UTILS;C:\BAT
```

The last step would be to move the actual ANTIVIR.EXE program—the potentially dangerous one—to a directory that's not specified in the DOS path. The benefit to all this is that when you or another user execute ANTIVIR at a DOS prompt, the operating system will find and run ANTIVIR.BAT, which has the internal Windows testing mechanism, rather than ANTIVIR.EXE, which could prove fatal to the system.

Conclusion

When engaged at the wrong time—like whenever Windows is running—some programs can ruin your day. It's not enough to verbally warn users or to make the DOS prompt a warning, because some people are just bent on self-destruction—or at least destruction of files and hard disk storage systems. Wrapping dangerous commands inside a batch file that tests for Windows' active state is a fail-safe solution that adds another layer of protective insulation to Windows 3.1x. ♦

Chris Morton is a long-time Windows professional. He's been writing about the subject for eight years, with credits including Cadence, Windows Shopper's Guide, Windows Tips & Secrets, Opening Windows, and Advanced Windows Tips & Secrets. He's also an accomplished Windows instructor and seminar leader, having presented advanced Windows sessions to thousands of information system professionals across the US.