

Controlling a Windows-Based Student Laboratory Keeping Windows Configurations Standard

by Chris Morton

Running over one dozen, Windows 3.0-based, stand-alone graphics stations in a school lab presents its own unique set of problems. Take a group of teenage and continuing adult students, introduce them to a graphical user interface that can be (too) easily customized and...WATCH OUT!

This is the scenario at the Graphic Arts program located at the Career Tech Center (CTC) in Traverse City, Michigan. Students arrive here from school districts in five counties. For the past three years, Dale VanHouzen has been teaching Aldus PageMaker, with some students also learning Corel Draw! and other Windows-based graphic arts programs.

From an instructor's standpoint, the problem of using the Windows interface in a classroom becomes an exercise in applying methods that allow students the freedom to learn and experiment, yet be able to easily reset the machines back to a known standard for the next group of users. This is especially true when at present, most CTC

students don't receive a basic understanding of DOS naming and file storage conventions prior to having access to the power of Windows 3.0.

(Crossover Technologies recommends Microsoft's "Learning MS-DOS," and their recently-intro-

duced "Productivity Pack for Windows" for new 3.0 users.)

With earlier versions of Windows, students couldn't change lab-specific default settings very much. Perhaps the worst problem was tracking down student-created files that they had saved to improper directories and wayward temporary files (*.TMP) that didn't stay put in the \TEMP directory (somehow disregarding the SET TEMP={drive\TEMP} statement in the AUTOEXEC.BAT file). To be safe, Dale had created a master backup of the entire system using FastBack Plus; he would then periodically restore each machine.

A set of error-detecting batch files were created for the lab by Crossover. These were selected through a character-based menu package and took care of copying compressed student work files back and forth from their individual floppy disks to a WORKAREA directory created on the hard disk for their use.

TIP: Aldus PageMaker, Corel Draw! and many other Windows applications have a nasty habit of leaving .TMP files on the hard disk whenever the system hangs. When that happens, you have to reboot without properly exiting the application software.

Over time, this can amount to several megabytes of storage space being wasted. If the SET TEMP = {drive}\TEMP environment variable statement is placed in the AUTOEXEC.BAT file, all Windows applications will leave such temporary files in one directory, facilitating cleanup. Adding this next line in the AUTOEXEC.BAT will eliminate these unwanted temporary files at the time of bootup:

```
IF EXIST {drive}\TEMP\*.TMP  
DEL {drive}\TEMP\*.TMP > NUL
```

Late last year, the character-based menu software was put aside permanently in favor of running each system with the new graphical user interface (GUI) supplied by Microsoft's latest release of Windows. With 3.0 installed, it became apparent that Program Groups (*.GRP), individual Program Items (software applications) and the overall appearance of the Program Manager interface could be too easily customized (or deleted!) by students, quickly resulting in a total lack of standardization between lab computers. Dale's time was already at a premium; having to rely on repeated manual backup and restore procedures to reset each computer was out of the question.

The solution was found in PROGMAN.INI, a separate Program Manager initialization file that Microsoft programmers were thoughtful enough to include in the new Windows release. This single file controls all the display criteria for the individual Program Groupings that appear on screen when Program Manager is activated. Since this is a simple ASCII file, try TYPEing PROGMAN.INI out to view its contents.

TIP: You may create different PROGMAN.{ext} files for different users or different functions. At



Graphics Arts students at the Career Tech Center.

Chris Morton is a PC graphics specialist with Crossover Technologies, Inc., of Traverse City, Michigan.

home, I've created a version for games only, so as not to confuse younger users with the overwhelming choices of my full-blown Program Manager. To do this, first copy your master PROGMAN.INI to something like PROGMAN.ORG. Then go ahead and create the new version. Existing Program Items may be moved between Groups by dragging them with the mouse; new Items and Groups are created using the FILE - NEW series of commands from within Program Manager. Both Items and Groups are deleted using the Del key. After you've created your new version, call up File Manager and copy the newly-edited PROGMAN.INI to {PROGMAN.new}. Lastly, create two batch files to copy either of your custom PROGMAN files to PROGMAN.INI just before issuing the WIN command.

Example:

```

@ECHO OFF

C:
CD\WINDOWS
COPY PROGMAN.ENT PROGMAN.INI > NUL
WIN
  
```

The > NUL at the end of line #4 simply prevents the message, "1 file copied", from being displayed. For regular use, line #4 of my other batch file reads:

```

COPY PROGMAN.ORG PROGMAN.INI > NUL
  
```

Because the CTC lab presents a situation where all default configuration files could be tampered with, knowingly or unknowingly, Crossover took this idea several steps further. In the case of SYSTEM.INI, a complete Windows reinstallation or backup restoration would be required if the original file could not be unerased or the corresponding *.BAK file couldn't be utilized. For each event, time is wasted restoring the machine to its original configuration.

Once Dale was satisfied with the Program Groupings and Items on one system, we compressed all the *.INI and *.GRP files into a master settings file using Phil Katz' "PKZIP" shareware program (commonly available on most BBSs). Now each time a lab system is booted up, this master ZIP file is uncompressed using the overwrite mode (-O) to the \WINDOWS directory:

```

CD \WINDOWS
PKUNZIP -O WINSET.ZIP > NUL
  
```

Aldus PageMaker and Corel Draw! also have their own configuration files that students can change too easily. Dale likes to maintain certain default settings for each, so as AUTOEXEC.BAT

runs, master versions of the corresponding files are copied to PM.CNF and CDCONFIG.SYS, respectively.

Now regardless of what the students do within Program Manager, the systems maintain their standardized, custom lab configuration. Another batch file, entitled RESET.BAT, contains just the command lines that pertain to these settings and can be issued at any time without rebooting, should someone really get carried away.

To provide access to system tools that could be misused by those lacking adequate knowledge of their proper usage, a separate Program Group was created for instructor-use only. Here's where icons controlling access to Norton Utilities, Control Panel and other system functions reside (see Figure 1). At present, there's no way to prevent unauthorized access, so we simply minimized this group to icon form and hid it behind the other groups already on screen when we saved the master copy of PROGMAN.INI.

File Manager was also part of this "Instructor" group until just recently; the LOAD= line of the WIN.INI has now been changed to bring File Manager (WINFILE.EXE) up as an icon in the lower portion of the screen as Windows is loaded. To demonstrate how files are manipulated using this program, Crossover used the Windows Recorder feature to create a functional macro that copies files from a student's floppy to the WORKAREA directory on the hard disk that is set aside for their use. A second macro reverses the process; both are linked to icons in a "Utility" group for student access. After students see firsthand how File Manager is used on the CTC lab systems, it's hoped they'll have a better understanding of what they're doing as they use it manually for this same purpose during successive class sessions.

Those who access the WINDOW menu pull-down from within Program Manager may stumble across the "Instructor" group choice and decide to investigate, but we'll live with this until Microsoft adds password protection to Program Groups in a future release of Windows. Heck, maybe the students will learn something! ♦



Figure 1. System control programs are hidden in an "instructor" group.