WINCMD - the Windows(tm) Command Line Interpreter

This is the *Ultimate* in command line processors! It is a pre-release (shareware) program that performs similar to the COMMAND.COM program under MS-DOS®, with which we are all so familiar. Nearly every common DOS function has been included (in some manner) within the Command Line Interpreter. Because this is a *PRE-RELEASE* version, not all of the functions are complete at this time. Most of the common functions are either performed internally, or are 'trapped' to prevent their use under Windows(tm) (functions such as SHARE, CHKDSK, etc.) with appropriate warning messages. Use of such programs would normally do damage to file or memory structures. Other functions will eventually be incorporated (FORMAT, VOL, LABEL, EDIT, etc.), but for now the DOS 'external' programs are run in response to these commands. Finally, special *ENHANCEMENTS* (such as TASKLIST, KILLTASK, background COPY, 'SET' improvements, etc.) allow the Command Line Interpreter to take advantage of the multi-tasking environment available under Windows(tm), or add desirable functionality to an already useful DOS command. Some of these enhancements are still 'stubbed out'; executing these functions will generate an information message that the 'command is not (yet) supported.' Watch for future releases!

If you like this program, you may register it for $20. Registration entitles you to (when available) one upgrade, plus some additional software which I think you might like (such as a 'Windows(tm) Scheduler' which runs programs on a given day at a given time). Note that the Windows(tm) Command Line Interpreter is a PRE-RELEASE version, and may contain operational 'bugs' or inconsistencies which could possibly (but not likely - I test very thoroughly) lead to loss of data or equipment damage or any other loss or damage. Basically this is a disclaimer saying "Use it at your own risk". There are no warantees, guarantees, or legal liabilities on the part of the programmer/developer (me) on this version.

(as of 11/10/91)
Registration and other correspondence can be sent to:

Robert E. Frazier
7414 Mesa College Drive #32
San Diego, CA 92111

Note that this is an apartment, and I won't live there forever. So, if you can't reach me at this address, you can try via COMPUSERVE MAIL. My Compuserve ID is 70172,177 and I am on often enough to make this means of correspondence reliable. If possible, please use COMPUSERVE for any bug reports or suggestions for improvement. I also accept creative ideas (hint hint).

ENHANCEMENTS

There are a *whole bunch* of enhancements in this program over the standard DOS functions they perform. Below is a list of some of the more (popular?) likely-to-be-used features - enjoy!

1) RUNNING WINDOWS APPLICATIONS FROM THE COMMAND LINE! This is probably *THE MOST* practical of all of the features! Not only that, *but* you can also put WINDOWS applications into BATCH FILES and then run the BATCH FILES! Note that applications spawn *immediately* when the application name is entered on the command line, or from a batch file. To account for this, the 'IF' command has also been enhanced (note item #5 below).

2) Background COPY operations! In an attempt to create true multi-threaded operations I have discovered a *really* nice way to perform background tasks under Windows(tm). So, I have included the BACKGROUND COPY feature in WINCMD to help gain popularity of this highly un-

utilized capability that Windows(tm) can *EASILY* accomplish (I also have a multi-thread API for Windows 3.0 which works under similar principles that is currently under developement...). SO, whenever the COPY command is entered from the command line (or a batch file) the COPY QUEUE gets all of the applicable file names added to it. For BATCH files: note that the 'COPYING' environment variable gets assigned a value of 'TRUE' when copying! Use the 'IF' statement to test when this is complete (you will want to do this anyway before exiting, although the EXIT command will give you a dialog box if COPY operations are still in progress).

3) DIR /AP and DIR /X - these two features allow you to view file attributes for program files! The DIR function attempts to determine what type of file it is, and if it's a program file (has an EXE header) or has an appropriate extension (.COM,.PIF,.BAT,.CMD) the 'Type' column contains the appropriate "id string" (type code in '<>', similar to '<DIR>'). Using '/AP' will only display those files with a non-blank id string. Try it! The '/X' command displays all files, and includes id strings for appropriate files. Note that the DIR function will open each file for READ/SHARE_DENY_NONE access in the process of determining the type, so sharing errors could potentially cause an '<ERR>' type to appear.

4) TASKLIST, KILLTASK, CLOSETASK - These 3 functions are Windows(tm) specific functions which allow the user to perform low-level task manipulation. The 'TASKLIST' function does not base its task information on top-level windows - it goes right into the *GUTS* of the KERNEL and grabs the master task list. This allows you to see ALL tasks which are running (not just those with windows!). KILLTASK and CLOSETASK represent two different methods of remotely ending a task. KILLTASK sends a 'WM_QUIT' message, which essentially blasts the program out of existence. Most likely a function terminated in this manner will NOT perform any necessary cleanup - it just QUITS! A safer (and gentler) way of terminating a task is the CLOSETASK command. This (essentially) does an ALT-F4 on each window owned by a task. Almost every task can be safely terminated in this manner, except for WINOLDAP (DOS) tasks, which don't do well because under Windows 3.0 you cannot ALT-F4 a WINOLDAP task - so when I do it anyway it *chokes* and causes minor problems - in the next release I will try to provide a means of handling this problem.

5) "IF ISTASK handle command line" - in addition to the other 'if' commands (note that ERRORLEVEL cannot be supported - see the on-line HELP for IF for more information) I have included the 'ISTASK' keyword to allow you to validate the existence of a task handle. Whenever a task is successfully run the 'TASK_ID' environment variable is assigned the task handle! Otherwise, it's BLANK (similar to "SET TASK_ID="). Note the following segment of batch commands:

```
  REM ** MINIMIZE COMMAND INTERPRETER WINDOW **
  IF NOT "%WINDOW_STATE%"=="MINIMIZED" MIN
  REM ** RUN 'MYTASK.EXE' BY SEARCHING PATH **
  MYTASK.EXE
  IF "%TASK_ID%"=="" GOTO IT_FAILED
  REM ** WAIT UNTIL TASK COMPLETES BEFORE CONTINUING **
  :WAIT_FOR_TASK
  IF ISTASK %TASK_ID% GOTO WAIT_FOR_TASK
  REM ** note that if this is towards the end of a  **
  REM ** relatively large batch file that the file  **
  REM ** currently gets read from the start to find **
  REM ** the label 'WAIT_FOR_TASK', so it's best to **
  REM ** include such loops within the first 4k of  **
  REM ** the file!                                  **
```

As part of the 'GOTO' loop, this function will automatically yield to other applications for each line it executes.

6) The MIN and MAX functions! These toggle MINIMIZED and MAXIMIZED window states. Feel free to experiment with these. Also, whatever state the window is in is stored in the environment variable 'WINDOW_STATE', which will be one of the following: "MINIMIZED", "MAXIMIZED", or "NORMAL".   You can use an IF statement to check for one of these 3 (or NOT one of these 3) and take appropriate actions.

7) The EXIT function has a parameter - well, a SPECIAL parameter. To end the Windows(tm) session, just use "EXIT WINDOWS" in place of "EXIT" - you will see a dialog box informing you of this, and you will have opportunity to cancel (just like with PROGRAM MANAGER). If an application refuses to exit, then you will get an appropriate message for that as well. If WINCMD is loaded as the SHELL application via the "SHELL=WINCMD.EXE" line in the SYSTEM.INI file (instead of "SHELL=PROGMAN.EXE") then if you exit the 'SHELL' copy of WINCMD you will get this message. Note that you can still run other copies of WINCMD. In a future release I will probably provide some means of determining which one is the 'SHELL'.

<u>KNOWN BUGS AND OTHER ANOMOLIES</u>

Well, there are a *small* number of problems/bugs which are probably worth noting... here goes:

1) MEM /C is not currently supported under Windows(tm) 3.0 (sorry). I having some difficulty getting the methods down for obtaining memory information while within Windows(tm) 3.0, and I wanted to get the product released. Therefore a warning message will appear concerning a 'DLL' which cannot be found when you attempt to use this command. It should be working when release 1.0 is ready.

2) COPY operations stop during command line editing - this is due to the dialog box that appears when you press F3. In order to make COPY operations continue I have to make this a MODELESS dialog box, and right now that would require too much additional code. Hopefully there won't be too many problems due to this condition.

3) 'FOR' is not supported. Again, I am trying to get the 'pre-release' out, and each time I add functionality the date keeps getting extended further and further. You've got to draw the line somewhere, and here it is.

4) No 'Windows(tm)' style HELP file. There *is* an on-line HELP facility which simulates MS-DOS® 5.0's HELP command, and it prints within the display window for the Command Line Interpreter.

5) Ctrl-C, Ctrl-Break, Ctrl-S, and Ctrl-Q are only partially supported. Functions such as 'TYPE' include Ctrl-S and Ctrl-Q support, and batch files may be interrupted via Ctrl-Break. However, this is the extent of the support (Sorry). Full support should be available in the regular release.

6) COPY /B and /A switches have no effect. Again, this is superficial functionality, and for the most part will have little effect on reality. You can put the switches in the command line, but it won't affect anything. It does BINARY copies by default.

7) COPY file1+file2+file3[etc.] doesn't perform correctly. If you try this, you'll get a UAE, so please don't try it.

8) COPY to the CON device produces unpredictable results. Please don't do it. Use 'TYPE' instead. Also, COPY from the CON device doesn't work. Please don't do this, either.

9) Using KILLTASK on certain tasks can produce undesirable results. If you were to use KILLTASK on the PROGMAN task handle it would terminate, but you wouldn't get an 'Exit Windows' message box. This can be good, and it can be bad. (Incidentally, if you terminate PROGMAN, the Windows(tm) Command Line Interpreter does *NOT* become the new shell!). Also, terminating a WINOLDAP (DOS) program can be disastrous!! It results in a *TIME BOMB* which goes off when it wants to. Please don't try it. I will put protection against this in a later release. Incidentally, 'CLOSETASK' also shouldn't be used on WINOLDAP tasks either... it doesn't behave as expected, and keeps additional DOS apps from running.

10) There is no AUTOEXEC file for the Windows(tm) Command Line Interpreter when used as the SHELL (SYSTEM.INI SHELL=WINCMD.EXE instead of SHELL=PROGMAN.EXE). However, you *could* (theoretically) include an argument on the command line when running WINCMD, and this argument will become the first command executed - this could (theoretically) be a batch file, similar to an 'AUTOEXEC.BAT ' equivalent.

11) The 'File' menu doesn't do anything. Actually, it's there because 'Exit' is in there for compatibility. The remaining (grayed) menu choices are reserved for future use.

12) Typing in a non-executable file (such as a WRITE file with a .WRI extension) does NOT run the associated program! This is against what's documented, but I haven't added the code necessary to perform this (hence the 'anomoly' listed here).

14) The PRINT function is not supported. Again, I haven't added the code necessary to perform this task, and it relates to the previous one so when I add it I add this one too. Next release, ok?

15) BLINKING text doesn't work (ANSI). Well, Windows(tm) doesn't really support BLINKING text, so I don't either...

Well I can't think of any more. If any of you out there find anything, please let me know prior to the next release, and I'll correct the problem.

R. E. Frazier - 11/10/91