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, interactive support via 'INPUT', the 'WAIT' command, 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) (Updated 5/11/92)
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.

(Added 5/11/92)

To get a list of other *special* environment variables type 'HELP VARIABLES' at the command prompt.

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'.

(features added 5/11/92)

8) A new command 'INPUT' has been added which allows a batch file to become interactive by requesting input from the user. The resulting input line is assigned to a specified environment variable. Optionally a prompt string can be included on the command line. In conjunction with this 2 'IF' options have been added: 'IF [not] ISYES' and 'IF [not] ISNO'. See the on-line help description for more information on these options, and the 'INPUT' function.

9) Another new command, 'WAIT', has been added. This command allows you to pause execution of command input and batch execution for either a specified amount of time, or until a specified time, date, or date/time combination.

10) The 'TREE' command has been supported using IBM Graphics characters to produce a graphical representation of your directory structure. The output format is (I think so, anyway) nicer than the one DOS 5 uses...

11) STDOUT re-direction has been partially supported. Appending files is still not possible (yet). No functions require STDIN redirection at this time, but will when PIPING is supported in the final release. The framework for these additonal capabilities is already in place.

12) A function 'PLAYSOUND' has been added which will play a '.WAV' file when the Multi-Media System API has been loaded (this system is currently part of Windows 3.1, and should be included in the Windows 3.0 Multi-Media extensions). If your

system is capable of playing '.WAV' files, you can use this function to play them either ASYNCHRONOUSLY or SYNCHRONOUSLY (see the On-Line HELP for a full description).

13) (some people don't like 13, so I'm excluding it)

14) The 'LISTOPEN' function (abbreviated "LO" if desired) will generate a list of open files either on the screen or to a re-directed file. The open information includes the Windows Program name and Task 'ID', if the file is associated with a Windows program.

15) Typing the name of a file which does *NOT* have an 'EXE', 'COM', 'BAT', or 'PIF' extension will cause WINCMD to search through the Registration Database (Windows 3.1 and later) and the 'EXTENSIONS' section of 'WIN.INI' (all releases) to find an appropriate program to open the file, based on the 'File Associate' information generated from FILE MANAGER.

16) Drag & Drop Support! Dragging a file from FILE MANAGER onto WINCMD and releasing the mouse button causes the program to behave as though you'd typed the file name in at the command prompt. NOTE: Only a SINGLE file may be dropped onto WINCMD at a time, for the current release.

17) CLOSETASK and KILLTASK can now use the MODULE name in place of the TASK/WINDOW ID. They may also be abbreviated "CT" and "KT", respectively. Typing "CLOSETASK NOTEPAD" will, naturally, close the NOTEPAD task UNLESS THERE ARE MORE THAN 1 "NOTEPAD" INSTANCES RUNNING - in this case WINCMD doesn't know which one you want to close/kill, so it prints a list of each (with their Main Window Captions) and then prompts you for the one to close/kill.

KNOWN BUGS AND OTHER ANOMOLIES

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), unless you obtain a copy of "TOOLHELP.DLL". I having some difficulty getting the methods down for obtaining memory information while within Windows(tm) 3.0, and I (frankly) have not received enough registrations to warrant spending the time on it. Therefore a warning message will appear concerning a 'DLL' which cannot be found (TOOLHELP) when you attempt to use this command. This is a "holdover" from when I was Beta Testing release 3.1, as I signed a Non-Disclosure Agreement and intended to honor it by not "spilling my guts" about the library name or its functionality.

2) COPY operations stop during command line editing - this is due to the dialog box that appears when you press F3. The code to rectify this problem will be added prior to release 1.0 .

3) 'FOR' is not (yet) supported. This will be supported, with *ENHANCEMENTS*, in release 1.0 .

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. Also, the 'WAIT' command and 'INPUT' command include Ctrl-Break/Ctrl-C support, and any command (like 'DEL *.*') which requests confirmation. Greater 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. BINARY copies are performed by default.

7) (Corrected in 2/17/92 update)

8) (Corrected in 2/17/92 update)

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.

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. Later it will be used for Object Linking and Embedding. The remaining (grayed) menu choices are reserved for future use.

12) (Corrected in 5/11/92 update)

13) (Missing again - some people don't like "13")

14) The PRINT function is not yet supported. It will be supported in the 1.0 release in conjunction with Object Linking and Embedding (OLE).

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

16) STDOUT re-direction not implemented on the 'TYPE' function. This will be done in the final release.

17) STDOUT re-direction doesn't support '>>' (file append). Yet.

18) File output to the 'PRN' device goes through DOS, not the Print Manager. Later I will implement a check for PRN, LPT1, COM1, etc. in the same manner as the checks for 'CON' are performed. That way, any output expected to go to a

printer will go through the Print Manager.

19) Because WINDOWS(tm) has some (minor) problems with the SHARE program loaded, you *may* experience troubles running program files. If this is the case, you might try specifying the program extension. If this does not help, plz send me a bug report indicating whether or not SHARE was loaded, what you typed in, what program you were trying to run, the directory you (and it) were in, etc. (These problems have been reported to Microsoft).

20) Windows(tm) does not appear to maintain more than one 'active' drive/directory combination for a given task. It is possible to change the directory that WINCMD uses for 'inactive' drives by using the 'File Open' dialog box of other applications ('WRITE', for example) on a drive that WINCMD doesn't have selected. This is a known problem, and I will be attempting to rectify this issue in the final release, so that *each* running copy of WINCMD has its own 'isolated' environment, including 'non-default' drives.

21) Piping support not yet implemented.

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

Updated - 05/11/92

**** RELEASE 0.93 NOTES ****

Release 0.93 has been (released, that is) to the public in a 'preliminary' state because, frankly, I have not received enough licenses (a total of 4, actually, not counting copies for "promotional" purposes) to warrant spending a great deal of time on this program. Since I have mouths to feed and bills to pay (as well as my wife's student loans) I have not been able to complete many of the functions (like FOR, piping, etc.) that are normally inherent to DOS. I have, however, made bug-fixes, improvements to the interface, and added many 'Windows Specific' functions which include many that take advantage of the new 3.1 release. This will be the only upgrade planned for quite some time, unless the program's marketing status changes radically (maybe some large distributor will buy it(?)). As *SHAREWARE* it does little more than make my name known as a developer, and get me some interesting contacts (thanx, Joel). I have placed some *VERY* rare and useful capabilities into this release, such as WAIT, LISTOPEN, INPUT, TASKLIST, CLOSETASK, KILLTASK, Drag/Drop support, and many others. No doubt I will receive a flood of ideas on how to improve this program, many of which have been implemented here already. Unfortunately I have not had the time to devote to making this the 'major application' that it *could* be. Perhaps in the future I shall.
R. E. Frazier, 5/11/92