WCL Help

WCL is a command line processor for Microsoft Windows 3.1, and IBM Win-OS/2. There are several internal commands available in WCL under the headings listed below. A proper introduction to WCL is contained within the "Introduction" section below. If you are new to WCL please read that section first.

REGISTRATION FEE: £18.00 (U.K. Sterling; or see the "REGISTRATION" section below for the fee if paid in other currencies)

Many thanks to those who have registered WCL. If you have not registered your copy, please consider doing so;

Please see the REGISTRATION SITES section below for your local registration site.

Alphabetical List of Commands

Below is an alphabetical list of internal commands supported by WCL. They can all be found grouped under the various services listed in the index screen.

MD
NEWCOMMAND
PATH
PAUSE
PLAY
PRINT
PRINTSCR
PROMPT
PWD
RD
REN
RENAMEWIN
RESTART
REVEAL
RUNDOS
SAVE
SAY
SEEK
SENDMESSAGE
SET
SETDATE
SETTIME
SPAWN
SUM
SYSDIR
TIME
TIMER
TIMER-ON-DESKTOP
TYPE
TYPEWRITE
UNZIP
UPDATE
WINDIR
WINLOAD
WINRUN
WINSHELL

Directory Services

WCL supports a number of internal commands for directory services. The supported internal commands operate similarly to the DOS equivalents. Sometimes however, there may be minor variations. The directory commands are outlined below

ALIAS
CD
DIR
GOTO
HOME
MD
PWD
RD

DIR

List the files in the directory. This can take parameters. e.g "DIR A:*.EXE", "DIR C:\MSDOS", etc. If no parameter is supplied, then there will be a listing of the CURRENT directory. The list displays about 20 lines and then pauses for a key press - much like "DIR/P" in DOS.) By default, the directory listing is sorted, according to the names of the files, with sub-directories appearing first before files. You can afterward scroll up and down the file list window with the mouse, or by using the PgUp, PgDn, and arrow keys.

The DIR command takes other parameters to change the order of sorting, or the format of the file listing. To see the options available here, type "DIR /?" (note that there must be a space between the "DIR" and the "/?" parameter).

The Directory listing is sorted, with sub-directories appearing first before files. You can afterward scroll up and down the file list window with the mouse.

The default sorting is by NAME, but you can change the sort order by the /O<D,E,S> switch.

/OD = sort by date

/OE = sort by extension

/OS = sort by size

Note that the directories are ALWAYS sorted by name. These switches only apply to normal files.

Other switches are;

/W = use wide list format (valid only in BIGWCL)

/S = list matching files in all subdirectories

(cannot be used in addition to any other switch, files are NOT sorted at all, and the output cannot be redirected to a file or the printer)

NOTE: The DIR command shows in its first column the attributes of each file in the directory, enclosed within "<>"

H stands for Hidden

S stands for System File

R stands for Read Only

A stands for Archive (i.e, normal file)

DIR stands for Directory (i.e, this is a sub-directory)

N/A stands for "No Attribute"

If a file has more than one attribute, they are all listed e.g. <HRSA> for the DOS system files.

Like with the DOS equivalent, the output of the 'DIR' command can be redirected to a FILE or to the printer (LPT1) with ">"

e.g. DIR *.EXE > EXEDIR.TXT (output to a FILE)

DIR C:\WINDOWS > LPT1 (output to the PRINTER)

NOTE: If you use redirection, make sure there is a space both before and after the redirection sign ">" (as in the examples above). If you do not insert a space (e.g. "DIR *.EXE>EXEDIR.TXT") you will probably only get an error message.

The DIR window show can be RESIZED and MOVED, and there is nothing to stop you from having many DIR windows open. It's quite a straight- forward matter to ensure that only one copy of DIR is running, but I am convinced that there are good reasons for allowing multiple copies. You can compare the contents of two directories by having DIR windows of both of them on screen, for example.

MD

MD - Create a Directory (alternative command is MKDIR).

PWD

PWD - Show the current directory.

CD

CD - Change to a directory (alternative commands are CHDIR and CWD). Using this command you can also change to an ALIASed directory.

RD

RD - Remove a directory. This will only work if the directory is empty of files (alternative command is RMDIR)

GOTO

GOTO - Change to a directory for which an alias has been created in WCL.INI. You should create ALIASES under the "[Directories]" part of WCL.INI by using a text editor or by using the ALIAS command.

The GOTO command enables a user to change to long directory paths easily.

e.g. if the line "SYS=C:\WINDOWS\SYSTEM" exists in the "[Directories]" part of WCL.INI, typing the command "GOTO SYS" will cause WCL to change to C:\WINDOWS\SYSTEM.

ALIAS

ALIAS - VIEW the current list of Directory Aliases in WCL.INI (the first 20), or CREATE a new Directory Alias, or CHANGE an existing one. If the command is used without any parameter, a list of current Aliases is presented.

To CREATE a new alias, or CHANGE an existing one, use ALIAS <ALIAS-NAME> <DIRECTORY-PATH>

e.g. ALIAS BACKUP C:\DOCUMENTS\SECRET\BACKUP

If you want to create an ALIAS for the directory in which you are, you can use a dot (".") for the directory path, e.g.,

ALIAS THIS-DIR .

HOME

HOME - Change to the WCL directory (alternative commands are GOHOME and HOMEDIR).

File Services

WCL supports a number of internal commands for file services. The supported internal commands operate similarly to the DOS equivalents. Sometimes however, there may be minor variations. The file commands are outlined below

ADD
ATTRIB
COPY
COPYTREE
DECODE
DEL
DELTREE
ENCODE
HIDE
PRINT
REN
REVEAL
SEEK
TYPE
UNZIP

COPY

COPY - Copy a file or a number of files. This again is similar to the DOS Copy command and wildcards are allowed. You can copy to another drive/directory, etc., or to the printer "LPT1" (alternative command is CP).

NOTE: You can create an ASCII file with "COPY CON <FILENAME>", just as under DOS (e.g., "COPY CON LOADWCL.BAT"). This is useful for quick creation of TEXT FILES from within Windows without loading the NOTEPAD, or any other Text Editor. When the "COPY CON command is invoked, an Edit Window is opened for the text to be typed in. Each line is NUMBERED by WCL, so you can know how many lines you have left (a MAXIMUM of 100 lines of text is permitted, and each line cannot be more than 128 characters in length).

NOTE that the editor is a LINE EDITOR, just as in the DOS command line. Each linr must be terminated by a Carriage Return and you CANNOT go back to edit previous lines.

When the editing is complete, type a period or full stop (".") on a line by itself, or type "end" on a line by itself to finish. It is at this point that the file is written to Disk.

The Lines Numbers supplied at the edit screen by WCL will NOT be written into the file, neither will the "end" or the period "." which inform WCL that you have finished editing.

NOTE: That you can also copy a file to the printer. "LPT1" and "PRN" are the only printer ports supported.

e.g. "COPY COMMANDS.SUM LPT1"

This will cause the file COMMANDS.SUM to be printed.

The COPY function tries to ensure that there is enough space on the destination drive for the files to be copied, on a file-by-file basis. If there is insufficient space for a file, there will be an error message to that effect, but the function will then proceed to try and copy any other file listed for copying. This is better than DOS in that DOS terminates the COPY function when there is insufficient space for ANY file, even if there are smaller files that will fit into the the target drive. WCL will copy these smaller ones.

The COPY function also tries to verify that the actual number of bytes copied are equal to the size of each Source file. If there is any discrepancy in the sizes of the copied file and it's copy, there is an error message informing you of this, and the copy is deleted.

DEL

DEL - DELete a file or a number of files. This again is similar to the DOS equivalent (alternative command is RM).

COPYTREE

COPYTREE - This command attempts to copy a file specification in a given diretory tree. This includes all the files matching the required specifications in that directory, and in its subdirectories.

The command attempts to re-create the directory tree structure of the SOURCE directory on the TARGET drive/directory. If a particular sub-directory in the tree cannot be created for some reason, the files that belong there will be copied into the root target directory.

If there is any problem with copying any file, the process will abort.

This command has certain restrictions. If it is used on the ROOT directory of any drive, it will NOT re-create the directory tree structure on the target drive/directory. It will just copy all the matching files into the target directory - period.

The syntax for the command is

COPYTREE <FILESPEC> <TARGET DIRECTORY>

If no directory path is supplied for the "filespec" then it will assume that the directory tree to be copied is the current directory.

Examples;

1. COPYTREE C:\WP*.DOC E:\MYDOCS

Means copy all the .DOC files in C:\WP and ALL its sub-directories, to E:\MYDOCS, re-creating the directory structure of C:\WP under the directory E:\MYDOCS.

2. COPYTREE *.* G:\BACKUP

Means copy all the files in the current directory and ALL its sub-directories, to G:\BACKUP, re-creating the directory structure of the current directory under the directory G:\BACKUP.

DELTREE

DELTREE - This command attempts to delete ALL the files in a given diretory tree. This includes all the files in that directory, and in its subdirectories. It then attempts to erase all the subdirectories in that directory tree. An alternative command is NUKE.

NOTE:

I have implemented this command against my better judgment, because users demanded it. I do not think that deleting files should be made easy, since recovering them again may be impossible. In my opinion, DELTREE is a command that is best left well alone. If anybody proceeds to use it, I am cannot accept any responsibility for any loss of data that may ensue.

Because of the drastic nature of what DELTREE does, I have imposed some limitations;

1. You will be asked to confirm TWICE that you wish to proceed.

2. You must supply the name of a valid directory to the command. Just typing "DELTREE" will be rejected. You need to type something like "DELTREE D:\JUNKMAIL"

3. The command will reject any attempt to apply it to the ROOT directory of any drive.

e.g., "DELTREE \"

or "DELTREE C:\"

these will NOT be accepted.

4. If there is any problem at all with deleting any file, then the process will abort.

REN

REN - RENname a file. This can be used on only one file at a time (alternative command is MV).

TYPE

(display the contents of an ASCII file, 20 lines at a time. Note that you should not put the "<" character before the file name (unlike DOS).

e.g TYPE MYFILE.TXT

or MORE HELLO.DOC (not "MORE<HELLO.DOC")

HIDE

HIDE - Hide a file or a number of files by setting their attributes to "hidden" (alternative command is CONCEAL).

REVEAL

REVEAL - Restore a file or a number of files from "hidden" to normal (alternative command is UNHIDE).

SEEK

SEEK - Try to locate a file or a group of files which match the specified filespec. The whole drive is searched for the files, and any matches found are listed. This command accepts wildcard characters. An alternative command is LOCATE.

e.g. "LOCATE WINWORD.EXE" - will look for all occurences of WINWORD.EXE.

"SEEK WP*.*" - will look for files matching this specification.

This command can now take an extra parameter ("/DELETE") after the file specification. This is useful for getting rid of files of a particular specification. The deletion takes place in the current directory tree only. If you want to use it to delete a file spefication through out the whole drive, you have to run this from the ROOT directory.

Use this parameter with care! You are only given ONE warning.

Example;

SEEK *.BAK /DELETE

This will seek for all files with a *.BAK extension, in the CURRENT directory tree, and delete any matching ones.

If the file specification is "*.*", the /DELETE parameter will NOT be accepted.

ADD

ADD - Add the contents of one file to another. The file that you want to ADD TO is to be specified LAST, and the the file that you want to add to it is to be specified FIRST (alternative command is CONCAT).

The syntax is thus;

"ADD <File to Add> <File Added To>"

e.g. "ADD SECOND.TXT FIRST.TXT"

This will append or add the contents of SECOND.TXT to FIRST.TXT. This means that after the operation, the file FIRST.TXT will now contain both the original contents of FIRST.TXT, with the contents of SECOND.TXT.

If for Example, FIRST.TXT originally contained "ABC" and SECOND.TXT originally contained "DEF", after the ADD operation, FIRST.TXT will now contain "ABCDEF". The contents of SECOND.TXT remain unchanged.

Always remember that the SECOND file to be specified is the file that will be ADDED to and that what will be added to it are the contents of the FIRST file to be specified.

Note: You CANNOT use wildcards in this command.

PRINT

PRINT - Print a file, i.e., send it to the printer "LPT1" (alternative command are LPR and LPT).

ATTRIB

ATTRIB - VIEW and/or CHANGE the attributes of a file or a group of files. To VIEW the attributes of a file, use ATTRIB <FILENAME> You cannot use wildcards if the ATTRIB command is used in this way.

To CHANGE the attributes of a file or files, use ATTRIB <ATTRIBUTES> <FILESPECS> You can use wildcards when the ATTRIB command is used in this way.

The ATTRIBUTES are represented by

R for READ ONLY;

S for SYSTEM FILE;

H for HIDDEN;

A for ARCHIVE.

You turn them ON or OFF by supplying a plus (+) or minus (-) AFTER the attribtutes.

e.g ATTRIB +RH HIDDEN.DOC (set HIDDEN.DOC to Read Only and Hidden)

e.g. ATTRIB -RS +HA SYSTEM.DOC (set SYSTEM.DOC to Hidden and Archive, and remove the Read Only and System settings)

ENCODE

ENCODE - This command is a Windows implementation of the UU-ENCODE function. Use it to UUencode a file. The command takes only one parameter - the name of the file to UUencode. The encoded file is given a.UUE extension.

eg

ENCODE THISPROG.ZIP - UUencodes THISPROG.ZIP to THISPROG.UUE.

See also;

DECODE

DECODE

DECODE - This command is a Windows implementation of the UU-DECODE function. Use it to UUdecode a file. The command takes only one parameter - the name of the file to UUdecode. The name for the decoded file is retrieved from the UUencoded file. If the filename supplied as a parameter is supplied without any extension, the extension .UUE is assumed. Note that this function is very slow.

eg

DECODE THISPROG - looks for THISPROG.UUE to decode

See also;

ENCODE

UNZIP

This command is designed to extract files from ZIP archives which have been produced by PKUNZIP(tm) or other ZIP utilities. The command can handle most forms of compression used by ZIP archivers, and can reproduce any directory structure found in the ZIP file, although this feature is turned OFF by default (you have to use the -D parameter to turn it on).

This command requires only one parameter - the name of the ZIP file to be processed. If no extension is supplied, a .ZIP extension is assumed. You can use wildcard characters here.

You can optionally supply a TARGET directory for the files to be UNZIPped into. If no target directory is supplied, the files are extracter into the CURRENT directory.

The UNZIP command can take further optional parameters, and these parameters can be in ANY order;

-V = VIEW the contents of the ZIP archive.

-D = Restore the DIRECTORY structure found in the ZIP archive

-F=<filespecs> = process only the specified filespecs. Note that there should be NO space between the "=" sign and the file specifications.

NOTE that the -D and -V parameters cannot be used together, and that only one copy of the program can be unzipping files at any particular time.

EXAMPLES:

UNZIP -V WINCMD73

- view the contents of wincmd73.zip

UNZIP -D WINCMD73

- extract the files in the ZIP file wincmd73.zip into the current directory, and restore any DIRECTORY structure found in that ZIP file.

UNZIP -D -F=*.EXE WINCMD73

- extract all files with the .EXE extension from the ZIP file wincmd73.zip into the current directory, and restore any DIRECTORY structure found in that ZIP file.

UNZIP WINCMD73 C:\WCL

- extract the files in wincmd73.zip to the directory C:\WCL - do not restore any directory structure in the ZIP file.

UNZIP -D WINCMD73.ZIP C:\WCL

- extract the files in wincmd73.zip to the directory C:\WCL - and restore any directory structure in the ZIP file.

Miscellaneous Services

WCL offers a number of services which do not readily fall under any of the previous headings. Thus they are all grouped under the miscellaneous services heading. The miscellaneous commands are outlined below

SAY
HELP
SUM
FREE
BEEP
PLAY
DOS
TYPEWRITE

SAY

SAY - Show a Dialog Box displaying whatever is typed after this command.

PLAY

PLAY - Play back any .WAV sound file. For example, to play the sound file CHIMES.WAV in the windows directory, type "PLAY CHIMES.WAV". Note that Windows requires a sound card to be installed before sound files can be played. If no sound card is installed, this command will produce no result. Alternative command is SOUND.

ABOUT

ABOUT - Show information about WCL (alternative commands are VER and ID).

HELP

HELP- Load this help screen (alternative commands are H and ?).

SUM

SUM - Load the Windows Notepad program with WCL.SUM. NOTE that WCL.SUM is a plain ASCII text file. (alternative commands are H2, HELP2, and ??).

FREE

FREE - Show the amount of free space on the drive which you specify after this command (e.g. FREE C: or FREE A:).

BEEP

BEEP - Make the annoying beep sound.

DOS

DOS - Open a DOS Shell. You return to Windows by typing "exit" (alternative command is SHELL).

TYPEWRITE

TYPEWRITE - This takes WCL into "Type Writer Mode". You are presented with an Edit Window wherein you can type text. When you press <ENTER> the LINE of text is sent to the printer ("PRN"). This command thus turns your Windows and Printer into a pretend Electric Typewriter. You can type as many lines of text as you wish, but bear these in mind;

[a] Each line must terminate with a carriage return

[b] Each line must not be more than 78 characters long

[c] You can have empty lines, just by pressing <ENTER>

[d] This command will NOT work properly with Page Printers (i.e., Laser printers). This is because lasers print one page at a time, and not line by line like dot matrix, inkjet, and daisy wheel printers.

To EXIT from typewriter mode, just type "END" on a line by itself, or a full stop "." ("period" in American) on a line by itself.

System Services

WCL offers a number of system services. Some of the commands are named like some internal DOS commands, but there are several which are unique to WCL. The system commands are outlined below.

In addition to the system services listed below, WCL allows you to use an "AUTOEXEC" batch file. This file is read everytime Windows is loaded, but only if WCL is your Windows Shell. This file should be called AUTOEXEC.CBF and should reside in your WCL directory, or in any directory which is in the DOS path. This file is treated as any normal WCL batch (.CBF) file, and should contain only commands that you wish to run EVERYTIME Windows is loaded. Note that batch commands may behave strangely because of the re-entrant nature of Windows and Win-OS2.

Please NEVER use WCL batch files in situations where things depend on commands being executed in a certain order. There is no way of telling the order in which the commands in your WCL batch file will be executed by Windows. Please note this warning.

WINRUN
WINSHELL

BACKUPTHEINIS

BACKUPTHEINIS - Make backup copies of the WIN.INI and SYSTEM.INI files. WIN.INI is backed up as WIN.WCL and WIN2.WCL, and SYSTEM.INI is backed up as SYSTEM.WCL and SYSTEM2.WCL. Alternative command is BACKINI.

CHANGE

CHANGE - Load the WIN.INI, SYSTEM.INI, and WCL.INI files into the Windows Notepad program for editing (alternative commands are CHG and CONFIGURE).

CHG2

CHG2 - Load WCL.INI into the Windows Notepad program for editing (alternative command is WCLINI.).

COLOR

COLOR - Change the color settings for text-color, text-background, and wallpaper, in WCL.INI. The command takes 3 parameters which determine each of the color settings. The settings that you specify are written to WCL.INI and will take effect when next you run WCL. The available colors are discussed in relation to the INI file settings, above. Alternative command is SETCOLOR.

e.g., COLOR black white white

GETCOLOR

GETCOLOR - Show the current color settings for TEXT-COLOR, TEXT-BACKGROUND and WALLPAPER, in WCL.INI. Alternative command is GETCOLORS.

CLS

CLS - Clear the screen. In WCL, this does nothing. In BIGWCL, it clears the screen, INCLUDING the contents of any scroll-back buffer.

WINSHELL

WINSHELL - Change the "SHELL=" setting in SYSTEM.INI. The command takes one parameter (i.e., the new Windows Shell). If no parameter is supplied, the name of the current Windows Shell is presented.

e.g. WINSHELL WCL.EXE

This changes the Windows Shell to WCL.EXE

WINRUN

WINRUN - Change the "RUN=" setting in WIN.INI. The command can take more than one parameter, each of them separated by spaces. The supplied parameters will replace the ones currently on the "RUN=" line. If all you want to do is to ADD extra programs to the "RUN=" line (as opposed to REPLACING the current one) then put a "+" sign BEFORE the first parameter

e.g., "WINRUN + DRWATSON.EXE WRITE.EXE"

this will ADD the two named programs to any one that is currently there.

(If no parameter is supplied, the current setting is presented).

To delete all the settings on the line, type "WINRUN NIL".

WINLOAD

WINLOAD - Change the "LOAD=" setting in WIN.INI. The command can take more than one parameter, and operates exactly like the WINRUN command.

EXIT

EXIT - Quit from WCL. If WCL is your Windows Shell, then this will quit from Windows. You will be invited to confirm that you do want to quit (alternative commands are QUIT or pressing ESC)

HALT

HALT - Same effect as with "EXIT" ,except that this command will exit Windows, whether or not WCL is your Windows Shell (alternative command is CLOSE).

SAVE

SAVE - Save the current state of the Windows desktop (the desktop is saved into a file called WCL.DSK in the Windows directory. This command also and saves the current WCL window co-ordinates and system settings (e.g., the current WCL prompt, the current Windows Shell, etc.) into the WCL.INI file. IF WCL is your Windows Shell, the saved Windows desktop will be restored when next you run WCL.

UPDATE

UPDATE - Update the WCL system settings by reading the WCL.INI file again. It is to be used only when you have changed the contents of the WCL.INI file in the current session and you want the changes to take effect immediately without EXITing and restarting WCL.

PATH

PATH - Display the current DOS path settings

TIME

TIME - Display the current time and date.

DATE

DATE - Display the current time and date.

SETTIME

SETTIME - Set the system time. Format is hh:mm:ss.
e.g. SETTIME 16:30:45

SETDATE

SETDATE - Set the system date. Format is dd:mm:yyyy.
e.g. SETDATE 22:04:1993

SET

SET - Show selected DOS and Windows environment settings. Note that this only SHOWS you the settings. You cannot use the SET command to change any of the environment variables.

PROMPT

PROMPT - Show the current WCL prompt, or change it to whatever is typed after this command.

e.g. PROMPT %

This gives you the UNIX percentage prompt

If you want a space to appear after your prompt, add a hash ("#") to the end of the prompt. e.g. "PROMPT FRED>#" ---- this will change the prompt to "FRED> ") To return the prompt to one that shows the current directory, type "PROMPT PG". Any prompt that is not PG is taken literally. No other "$" setting is supported. Please note this fact. Note also that you should NOT use the hash for "PG"

RESTART

RESTART - Shut down Windows and restart Windows again (valid only for Windows 3.1; alternative command is WIN).

COMMAND ALIASES

COMMAND ALIASES - Users can create an alias for any command, up to a MAXIMUM of 30 aliases. This has required a new "[commands]" section to be added to WCL.INI, just before the "[directories]" section. Unlike the directory aliases which require WCL.INI to be read each time, command aliases are loaded into memory only ONCE - when WCL is loaded. Thereafter, they are processed from memory, until you exit the current WCL session. This makes the feature as fast as internal WCL commands. But it also means that any new command alias that is created in the current WCL session will not be evaluated until the next time you run WCL (unless in cases where the new command alias was created by using the "NEWCOMMAND" command).

Restrictions are that the whole line on which the alias exists cannot be more than 79 characters in length, and that you have to create the command aliases by manually adding each new alias to the "[commands]" section in WCL.INI, or by using the new command "NEWCOMMAND" (see below).

Examples of manual entries in WCL.INI;

```
SYSDIR=DIR C:\WINDOWS\SYSTEM
CWCL=CD C:\WCL
CT=COPYTREE
```

Note that the command aliases are evaluated *before* internal and external commands. Thus, if you create an alias with the same name as a WCL internal command, it is that alias, instead of the WCL internal command that will be executed.

Note also that WCL interprets the aliased commands LITERALLY. Thus, it is the user's responsibility to ensure that the commands for which aliases are being created are correct and non-destructive.

Finally, do NOT create a command alias which involves the execution of a WCL batch file. Please note this point, as I cannot guarantee how the whole thing will be evaluated, if it involves the execution of a .CBF batch file.

NEWCOMMAND

NEWCOMMAND - This command is for the purpose of creating a new command alias (see above), without having to edit WCL.INI manually. This command inserts the new alias into WCL.INI, and retains it in memory for use during the current session. Note that if this command is used to replace a command alias which already exists in WCL.INI, the new one will not take effect until when you next run WCL. Use it only to create NEW command aliases.

Syntax

NEWCOMMAND <Alias Name> <Command>

Examples;

NEWCOMMAND BACKIT COPY *.DOC A:\
NEWCOMMAND CT COPYTREE

To delete a particular entry from WCL.INI, supply "NIL" as the command. Note that when you do this, the command alias WILL be deleted from WCL.INI, but it will still be active in memory, until when next you run WCL. So if you type LISTCOMMANDS to see the currently active command aliases, the one that has just been deleted will still be there. Please note this point.

LISTCOMMANDS

LISTCOMMANDS - produces a list of the first 20 command aliases.

SYSDIR

SYSDIR - Changes to the Windows SYSTEM directory.

WINDIR

SYSDIR - Changes to the Windows directory.

DOSKEY

DOSKEY - Use this command to enable or disable DOSKEY emulation during the current WCL session. Nothing is written to WCL.INI. This command takes one parameter (either "ON" or "OFF"). Note that using this command also affects support for assigning commands to the funtion keys. If the command is typed with no parameter, then you are just shown the current status of the DOSKEY emulation.

LISTWINS

LISTWINS - displays a list of all the currently active processes, with their process ID numbers. You need this information in order to use the RENAMEWIN or KILLPROG command. An alternative command is LW.

KILLPROG

KILLPROG - you can use this command to terminate any currently active process, by its process ID number. Get a list of all the currently active processes, with their process ID numbers by typing "LISTWINS". You then supply the process ID number of the program you wish to terminate, to KILLPROG. An alternative command is KP.

Note that once you supply an ID number to KILLPROG, the program which is represented by that ID number is terminated WITHOUT WARNING. Also, note that LISTWINS does not list some programs because I consider that it will be foolish to terminate them. Thus, if you type KILLPROG and then supply an arbitrary number, it may actually represent the ID of a running process, and that process WILL be terminated.

Therefore, use this command with care, and use it only with process ID numbers supplied by LISTWINS. I cannot accept any responsibility for wrongful use of this command (or, indeed, any other WCL command).

Example: KILLPROG 4309
This will terminate the program which has the ID number 4309, without warning.

RENAMEWIN

RENAMEWIN- you can use this command to rename the main window of any active process, by its process ID number. Get a list of all the currently active processes, with their process ID numbers by typing "LISTWINS". You then supply the process ID number of the program you wish to rename, to RENAMEWIN, and you also supply the new name that you wish to give to that window. The new name is active until you close down the program, or rename it again.

This command does NOT have a permanent effect, and so is quite safe. Alternative commands are RENW, and RENWIN.

Example: RENAMEWIN 4309 My Greatest Windows Hack
This will rename the main window of the program which has the ID number 4309, to "My Greatest Windows Hack", until you exit the program, or until you rename it again.

DO

DO - with regard to the way WCL runs external programs, some users have complained about the fact that WCL sometimes changes to the directory in which the executable for the program was found, before running it. This feature exists for many reasons (partly to do with Windows itself), but it can be circumvented. I have decided to keep that feature as the default, but to permit users to circumvent it. This is implemented when the user types "DO" before the name of the program to be executed.

If this is typed before the name of an external program then WCL will remain in the current directory while running that program, and will not change to the program's directory as it sometimes does. The only use of this command is as a prefix to whatever you would normally have typed. Thus if you would normally type;

```
PKUNZIP WINCMD*.ZIP
```

you would now type;

```
DO PKUNZIP WINCMD*.ZIP
```

Most people will never need to use this "DO" command, but it is there anyway. See also the RUNDOS, and SPAWN commands.

RUNDOS
SPAWN

RUNDOS

RUNDOS - Sometimes, users experience some problems with some DOS programs. A typical situation is a case where the user is trying to pass some parameters to a DOS program (e.g., to redirect output from the DOS program to a file). If you have persistent problems with running a particular DOS program through WCL, and nothing else works, try the RUNDOS command.

This is used by typing "RUNDOS" before the name of the program to be executed. What this does is to call your DOS command interpreter (i.e., the "COMSPEC" environment variable) and then pass to it anything that appears after the "RUNDOS" command. This is in effect equal to running the program from the DOS prompt. If you would normally type;

```
PKUNZIP -VN WINCMD73.ZIP > THIS.TXT
```

you would now type;

```
RUNDOS PKUNZIP -VN WINCMD73.ZIP > THIS.TXT
```

NOTE: you should use this command for DOS programs ONLY. Do NOT use it to run Windows programs - it will certainly not work. See also the DO, and SPAWN commands.

DO
SPAWN
DOS_Programs

KEY ASSIGNMENTS

FUNCTION KEYS (F1=F9, and F11-F12)

CONTROL KEYS (CTRL-A to CTRL-Y)

WCL supports the assigning commands to FUNCTION KEYS (F1 to F9, and F11 to F12), and the CONTROL KEY + AN ALPHABET (CTRL-A to CTRL-Y, but MINUS CTRL-C, CTRL-G, CTRL-H, CTRL-M, CTRL-T, and CTRL-X - these are reserved). Note that this feature is disabled if DOSKEY EMULATION is turned off.

The commands have to be assigned manually in KEYS.WCL, which is a plain ASCII file that can be edited with Notepad. Any command can be assigned to any of these keys, and any entry is executed IMMEDIATELY, EXACTLY as it appears in KEYS.WCL.

Examples:

F1=HELP

F2=

F3=*

F4=

CTRL-A=WPWIN STATUS.WP

CTRL-B=

CTRL-D=

..... and so on .....

This is how you assign commands to the function keys and the control keys. I have pre-set F1 for HELP, and F3 to bring up the last command (the asterisk means "fetch the last command").You can change these settings, and you can assign any command to any of the others (except F10). The commands are executed IMMEDIATELY after the key is pressed (ie you do not need to press ENTER). None of the control keys is preset.

Example: F2=COPY *.DOC A:\

With this setting, when you press the F2 key, all the files with a .DOC extension are copied to drive A:

A known problem with this feature is that sometimes you may have to press a function key twice, before the command is executed. I have so far been unable to find a way of remedying this situation. The problem does not exist with the control keys.

SPAWN

With regard to the way WCL runs DOS programs, some users have complained about the fact that WCL sometimes closes the DOS window before they can read the output from the window. This is actually a function of Windows, not WCL. However, I have provided an effective (if inelegant) way to bypass this.

This is implemented by prefixing the program with the "SPAWN" command. When the user types "SPAWN" before the name of a DOS program to be executed, WCL puts everything that appears after "SPAWN" into a batch file, and executes the file. When the file has finished executing, you are invited to press ENTER to return to WCL.

Example: SPAWN PKUNZIP -V *.ZIP | MORE

This will run PKUNZIP with a pipe into the MORE command. When PKUNZIP finishes the DOS window will still be open, and then you will be invited to press ENTER to return to WCL.

NOTE:

With this command, you can also get WCL to execute the COMMAND.COM version of internal DOS commands (eg DIR, PATH, SET, COPY) - instead of the WCL version.

e.g.

"DIR *.DOC" - this will execute WCL's own DIR command

but;

"SPAWN DIR *.DOC" - this will execute the COMMAND.COM version of DIR, in a separate DOS window.

NOTE: do NOT use this command to run Windows programs. It will most certainly NOT work. See also the DO and RUNDOS commands.

DO
RUNDOS
DOS_Programs

SENDMESSAGE

This command can be used to send "messages" to any active application through its process ID number. The process IDs are obtained by the "LISTWINS" command.

The messages that can be sent are "HIDE" (tell the process's main window to hide itself); "MAX" (tell the process's main window to maximise itself); "MIN" (tell the process's main window to minimise itself); and "NORMAL" (tell the process's main window to restore itself to the normal display).

It is the user's responsibility to make sure that the correct process is being sent the message.

LINE-EDITOR

LINE-EDITOR - Use this command to enable or disable the COMMAND-LINE EDITING functions during the current WCL session. Nothing is written to WCL.INI. This command takes one parameter (either "ON" or "OFF"). If the command is typed with no parameter, then you are just shown the current status of the LINE-EDITING function.

You can set the default value of this function in the "LINE-EDITOR=" line in WCL.INI. See more information in the "INI File Settings" section.

NOTE: if DOSKEY emulation is disabled, then the line-editing functions will not work.

TIMER

WCL now has a "TIMER" feature. This is enabled by the new "SHOW-TIMER=" setting in WCL.INI, or by typing "TIMER ON" at the WCL prompt. Note that this feature has a number of very serious limitations - so you might not want to use it at all.

What it does is to activate a very pathetic on-screen clock. It is pathetic because it has a number of problems;

[a] it stops being updated if the input focus changes from the WCL window (until the input focus is returned to the WCL window - if you are running under OS/2, and until the mouse is moved over the desktop or the WCL window if you are running under Windows);

The situation under OS/2 is so as to allow better multi-tasking when WCL is in the background.

[b] it is displayed inside the WCL window by default. This can cause a number of problems with untidy screen displays, when the PROMPT, or the commands being typed over-run the ticking clock, or when the WCL window is moved from one place to another.

The only other place I have managed to implement the display of the clock is on the Windows desktop (at the top right corner). You can use this option by setting the "TIMER-ON-DESKTOP=" line in WCL.INI to "ON" or "1", or by typing "TIMER-ON-DESKTOP ON" at the WCL command prompt.

The problems with displaying the clock on the Desktop are, [a] that the clock can be covered by other windows, and, [b] when WCL is closed, you can see still see the last clock display on the Windows desktop (you can get rid of this by maximising a window, and then restoring it to normal again).

[c] There are problems if you try to run FULL SCREEN DOS programs when the timer is ON - often you just get dumped unceremoniously to the DOS command prompt. Thus, if you are going to run FULL SCREEN DOS programs from WCL, then you should NOT enable the timer feature, or you should type "TIMER OFF" before running the program.

In short, this "TIMER" feature is NOT very good, and that is why it is turned OFF by default. If you find that it is NOT good enough for you to use, then you can forget that I have said that it exists, and leave it disabled. If you can live with its short-comings, then you can always turn it on. You can also turn it off for the current session by typing "TIMER OFF" at the WCL command prompt.

When I can improve on this feature, I will do so - but don't hold your breath. After all, a ticking clock is an un-necessary feature.

TIMER-ON-DESKTOP

This command can be used to state whether the on-screen clock (if enabled) is displayed on the Windows Desktop or not, and any changes made here last only for the current WCL session. See the "TIMER" command for fuller information.

TIMER

PRINTSCR

It is possible to print the contents of the WCL main windows. With BIGWCL, you use the command "PRINTSCR". If this command is run without any parameter, the contents of the window buffer are written into an ascii file called BIGWCL.SCR. You can however direct output to the printer by supplying "LPT1" or "PRN" as a parameter, and you can direct output to another ascii file by supplying another file name.

eg

PRINTSCR	- output to BIGWCL.SCR
PRINTSCR PRN	- ouput to the printer
PRINTSCR SCREEN.TXT	- output to SCREEN.TXT

This command does not work with the small windowed version (WCL.EXE), because there is nothing to print there. However, when you run commands like "DIR" from the small windowed version, a "Print Window" menu item is inserted into the directory listing window. Clicking on that menu item brings up a dialog that allows you to direct output to the printer, or to an ascii file called WCL.SCR.

Introduction

WCL contains too many features and commands to be fully summarised here. Please read this help file carefully.

Windows Command Line (WCL) is a command line interface program for Windows 3.x and Win-OS/2. The program simulates the infamous C:\> prompt of the DOS command line, but from within Windows, or while running as a "seamless" Windows application on the OS/2 Workplace Shell desktop. This is useful for those DOS hackers who find themselves having to use Windows for certain applications, or for people who want a very quick and easy way to multi-task Windows programs, either within Windows itself, or from the OS/2 desktop, or for those who like to have a command line window available at all times. From WCL, you can run all Windows, DOS and OS/2 (under OS/2 2.1) programs just by typing the program name, and pressing <ENTER> as you do would at the DOS prompt for DOS programs. When you run a program through WCL, the program's window becomes the Active Window. You can go back to WCL by clicking on any part of the WCL window that is visible to you, and then run other programs from there.

WCL consists of 3 main executables (1) WCL.EXE - the main executable (2) WCLDLL.EXE - library executable for WCL.EXE, and (3) BIGWCL.EXE - the "big" version of WCL.EXE, in the sense that [a] it's main window is big, and all the output is directed to that window, and [b] it is self-contained - unlike WCL.EXE, it does not require the library file WCLDLL.EXE.

WCL.EXE and BIGWCL.EXE are alternative forms of the main WCL program. There is nothing stopping you from using both of them, but the idea is that while some people will prefer WCL.EXE's small and unobtrusive main window, others may not like the fact that it uses popup windows for some of its output. The ONLY significant differences between WCL.EXE and BIGWCL.EXE lies in the size of their main windows, and the fact that all the output from BIGWCL.EXE is in the same main window.

OS/2 version 2.1:

WCL has been tested extensively under OS/2 version 2.1, and has been designed to detect that operating system and adjust itself thereto. It works beautifully under OS/2. If WCL is run from within OS/2 version 2.1, either as a "seamless" application (i.e., from the Workplace Shell desktop), or in a full screen Win-OS/2 session, you can run DOS, Windows and OS/2 programs from the WCL prompt.

OS/2 users should please NOTE that WCL will only run DOS and OS/2 programs under OS/2 in the same way that the Program Manager will run them, since WCL uses the same API calls. Some OS/2 programs will not run at all, if run from a Win-OS/2 session, and some DOS programs behave quite strangely if run from Win-OS/2.

PATH:

If the application you wish to run is not in a directory which is in DOS Path, you will have to supply the full path name (e.g. "C:\WPWIN\WPWIN", to run WordPerfect for Windows, if C:\WPWIN is not in the DOS Path). If the application is situated in a directory that is in the DOS Path, all you need do is type its name, and press <ENTER> (e.g. "WRITE" <ENTER>, to run Windows Write).

All Windows programs can be run from within WCL. This includes DOS programs for which a Windows .PIF file exists. Most DOS programs can also be run directly from

WCL without creating a PIF file for them. In this case, they will run in full screen mode.

Note that most internal DOS commands (i.e, those that are resident in COMMAND.COM) can NOT be run directly from WCL. However, a number of DOS-like commands are supported through built-in technology. Below is a list of them;

[1] CD or CHDIR (change directory)

[2] MD or MKDIR (create a new directory)

[3] RD or RMDIR (remove/delete a directory)

[4] DEL or ERASE (delete files. Wild cards are accepted)

[5] REN or RENAME (rename one file; Note - You CANNOT use wildcards!)

[6] COPY (copy files. Wild cards are accepted.

[7] TIME (show current system time)

[8] DATE (show current system date)

[9] SET (show SOME enviroment variables)

[10] PROMPT (Change the WCL prompt)

[11] TYPE or MORE (display the contents of an ASCII file)

[12] PRINT (print a file)

[13] DIR (list the files in the directory)

Apart from changing drives (e.g. "A:" to change to drive A or "D:" to change to drive D, etc.,) INTERNAL DOS commands different from those listed above cannot be directly run from WCL. Attempting to run them will either produce an error message from Windows, or lead to the DOS prompt being invoked through a DOS Shell.

External DOS commands (i.e, those which have their own .EXE, .COM, or .BAT files, e.g "FORMAT", "GWBASIC", "XCOPY", etc.) can normally be run directly from WCL. However, I would not attempt to run programs such as "CHKDSK" or programs which access the hardware directly (such as disk compressors) when in Windows. A lot of grief can result from this. Basically, any DOS program which can be used safely under Windows can be used safely in WCL since everything that WCL does is done through Windows API calls (i.e. Windows itself does all the actual processing. WCL only acts as a command line interface between you and Windows).

NOTE: the command "DO"

With regard to the way WCL runs external programs, some users have complained about the fact that WCL sometimes changes to the directory in which the executable for the program was found, before running it. This feature exists for many reasons (partly to do with Windows itself), but it can be circumvented. I have decided to keep that feature as the default, but to permit users to circumvent it. This is implemented when the user types "DO" before the name of the program to be executed.

If this is typed before the name of an external program then WCL will remain in the current directory while running that program, and will not change to the program's directory as it sometimes does. The only use of this command is as a prefix to whatever you would normally have typed. Thus if you would normally type;

```
PKUNZIP WINCMD*.ZIP
```

you would now type;

DO PKUNZIP WINCMD*.ZIP

Most people will never need to use this "DO" command, but it is there anyway.

You can dispense with having to type "DO" all the time by setting the "DO=" line in WCL.INI to "1", or "ON" (i.e., "DO=1"). With this setting, WCL will behave as if you have typed "DO" before the name of *every* program that you want to execute. This setting is NEW (with version 7.0), and is disabled by default.

If you still have problems with a DOS program and "DO" does not fix the problem, then try the "RUNDOS" command.

If there is any program which you should not run under Windows, then please do NOT attempt to run it via WCL.

Note that you can use the UNIX names of some of these commands, although they do not operate like the UNIX commands.

e.g.

CP for COPY

MV for RENAME

RM for DELETE

CWD for CHANGE DIRECTORY

LS for DIRectory listing

These commands operate more or less like their DOS equivalents, except that you cannot use wild cards in the RENAME function. For file copying, wildcards are accepted for SOURCE file specifications only. You cannot use wildcards in TARGET file specifications.

e.g

COPY *.DOC A:\MSDOS - This is valid

COPY *.DOC A:\MSDOS*.BAK - This is invalid.

REN *.DOC *.TXT - This is invalid

Both the COPY and DIR commands produce their own Windows on the screen. However, if you are running BIGWCL.EXE, all the output is in the main window. BIGWCL.EXE is almost an exact reproduction of the MSDOS prompt, within Windows/Win-OS/2.

Command_Line_History

WCL supports a limited form of command line history by keeping a record of the LAST 30 commands typed at the prompt. There are a number of commands for accessing the history function. They are enumerated below;

1. !! - (two exclamation marks) - this will execute the most recent command.
2. ! - (one exclamation mark) - If this is typed by itself, WCL will list the last 20 commands (each of them with a number) in a message box. When the message box is closed you are prompted for the number of the line that you want to execute. The command is then executed.

If you do not want to execute any of the listed commands, type 0 (zero) or just press <ENTER> If the single exclamation mark is followed by a space and then a number (e.g., ! 10), WCL fetches the command with that number (if any exists). Thus for example, "! 6" means fetch the sixth to the last command. You can use a hyphen instead of a space (e.g., "!-6")

3. LIST - Show a numbered listing of the last 20 commands typed at the WCL prompt (alternative command is HISTORY).
4. CLEAR - Clear the command line history list. This gets rid of all the entries present on the list of the last 20 commands. The list will then start to build from the scratch. It is a great way to stop prying eyes (e.g., the boss) from seeing what commands you have been typing all day.

DOSKEY EMULATION

WCL features support for a LIMITED emulation of the DOSKEY function of scrolling through a list of past commands (by using the arrow keys - LEFT or UP arrow keys, or CTRL-Z for ("up") and DOWN or RIGHT arrow keys, or CTRL-X (for "down")).

This feature is enabled by a setting "EMULATE-DOSKEY" in WCL.INI. A setting of "1" or "ON" enables this feature, and any other setting disables it (eg "EMULATE-DOSKEY=ON"). It can also be enabled or disabled temporarily within WCL by typing "DOSKEY ON" or "DOSKEY OFF".

One limitation is that you can only edit the commands by using the BACKSPACE key, unless the command-line editing function is turned on (see below).

FULL COMMAND-LINE EDITING

WCL now supports full command-line editing. This feature is optional, but it is turned on by default in WCL.INI (with the setting "LINE-EDITOR="). You can turn it off permanently by putting a "0" or "OFF" on that line. You can also turn command-line editing off and on temporarily by using the new command "LINE-EDITOR ON" (to turn it on) or "LINE-EDITOR OFF" (to turn it off).

If DOSKEY emulation is not enabled, command-line editing will not work. When command-line editing is enabled, you can use the left and right arrow keys, Backspace, Delete, Ins, Home, and End, to move about and edit the commands being entered, and also to edit the commands that have been brought up through DOSKEY scrolling. line editing function.

When command-line editing is enabled (the default setting in WCL.INI), the text being typed at the command line will scroll horizontally on the same line, and you can move left or right with the cursor keys. If the end of the prompt is less than 20 characters from the end of the WCL window, the cursor will move to the next line to

allow more room for the commands to be typed.

When command-line editing is turned off, then the text being typed will wrap onto the next line. In this case, you cannot go back to the previous line.

Batch_Files

WCL supports sequential processing of commands by allowing you to put commands into a BATCH FILE. This batch file must have ".CBF" as its extension. "CBF" stands for "Command Batch File".

e.g "COPYBAK.CBF"

Batch files can contain any command that WCL supports - ie. internal WCL commands, DOS .BAT, .EXE and .COM programs, Windows programs, and OS/2 programs. The file batch file must be in ASCII format, and each command must be on a separate line. Each batch file can be up to 30 lines. However, it cannot contain a reference to another WCL .CBF file.

Once set up, all you need to do is to type the name of the batch file at the WCL prompt. You do not need to type it's extension. With the example above, you only need to type "COPYBAK". WCL will then try to execute the commands in the file on a line-by-line basis. This may result in some interesting screen maneouvres as each program is given the input focus by Windows, and tries to display its messages and main window.

Remember that Windows programs do not have the whole PC to themselves, unlike DOS programs, so each Windows program will allow another to be immediately loaded after it, as soon as it sets up its main window. If the batch file contains a mixture of DOS and Windows programs, the screen maneuovres are yet more interesting. The import of this is that the processing of batch commands in Windows will not always be as you expect, if looked at from a DOS batch file point of view. This is due to the nature of Windows itself, and there isn't much that I can do about it.

NOTES:

Because of the multi-tasking functions in Windows, the NORMAL use of batch commands in situations where things depend on commands being executed in a certain order is impossible. This is because there is no way of telling the order in which the commands in your WCL batch file will be executed, since they will NORMALLY be executed in rapid succession. - i.e., unless you use the EXECWAIT, PAUSE, or DELAY commands.

If you MUST run the commands in the WCL batch file in a certain order, then you MUST use the command "EXECWAIT" - for EXTERNAL programs and commands, and either "PAUSE" or "DELAY" for WCL INTERNAL commands. problem.

NOTE also that WCL treats .CBF batch files as INTERNAL commands. This means that .CBF files will be executed in preference to any program of the same name, even if that program is in the current directory and the .CBF file is not. Thus for example, MYPROG.CBF will be executed in preference to MYPROG.EXE or MYPROG.COM. If you want the EXE or COM file to be executed instead, you need to type its EXTENSION as well.

EXECWAIT, or EW:

For use in running EXTERNAL commands through WCL batch files. What this command does is to cause WCL to WAIT until the program being run has terminated, before running the next command in the batch file.

To use EXECWAIT, all you need to do is to put it before the command that you would

normally type.

eg

EXECWAIT NOTEPAD TEST.TXT

EW WRITE MYDOC.WRI

EW PKUNZIP -V MYZIP.ZIP

PAUSE:

For use in running either EXTERNAL or INTERNAL commands through WCL batch files. Put PAUSE on the line following a command, and then WCL will pause for a key press before executing the next line in the batch file.

DELAY:

For use in running either EXTERNAL or INTERNAL commands through WCL batch files. Put DELAY on the line following a command, and WCL will stop for a bit, and then continue with the batch execution. "DELAY" can take one parameter - the number of SECONDS that WCL should wait for. If no parameter i supplied, WCL assumes a DELAY value of 1 second.

NOTE: there is an UPPER LIMIT of 300 (5 minutes) for this parameter. This number is arbitrary, and could be higher. However, I have decided to place it at this limit, as a safety margin. If enough people feel that I should remove this upper limit, then I will do so.

Examples;

DELAY	will wait for 1 second
DELAY 5	will wait for 5 seconds
DELAY 180	will wait for 180 seconds - 3 minutes
DELAY 1800	will wait for 300 seconds - 5 minutes

Commands in WCL batch files will be echoed on the screen before they are executed. If you want this behaviour to change, you need to put the line "ECHO OFF" at the beginning of each WCL .CBF batch file. You can turn the ECHO on again in the batch file by putting "ECHO ON" before the next command that you want to be echoed to the screen.

You can use "REM" in a WCL batch file. Any line that starts with a REM will not be executed. Note that there MUST be a space between the REM and any other text on the line.

You can pass parameters to your WCL batch files, just as in DOS. To do this, use "%1" as you would in DOS. This feature is still quite new, but it has worked well so far in my tests. You will have to experiment to see what works in this regard. Please use this only for the purpose of passing parameters to external programs. Do not use it for WCL's internal commands. Note: WCL can only process a maximum of 7 parameters in this way (i.e., %1, %2, and %3, etc., up to %7).

NOTE:

Please ensure that batch files do NOT have the same names as any DOS or Windows program file that you will call from the batch files. For example, if you will call KERMIT.EXE from you batch file, make sure that the batch file is not called KERMIT.CBF - if you do not heed this advice, you are SURE to get a SYSTEM CRASH when you try to run the batch file. Please note this warning.

AUTOEXEC.CBF

WCL supports an "AUTOEXEC" batch file for Windows/Win-OS/2. If WCL is your Windows Shell, you can put any commands that you wish to be executed every time Windows is started in a WCL batch file called AUTOEXEC.CBF. WCL will then run the commands in this file whenever Windows is started. If WCL is not your Windows Shell, the AUTOEXEC.CBF file will be ignored.

Example of a .CBF file's contents;

```
ECHO OFF
SAY     This is a test     .CBF file!
CD C:\WINDOWS
COPY *.INI A:\
DELAY 3
CD C:\DOCS\LETTERS
COPY *.LET A:\LETTERS
DELAY 2
SAY   I have finished the back ups
```

This file starts by printing a message that it is a test .CBF file. It then backs up all the .INI files in the Windows directory, and all the .LET files in the C:\DOS\LETTERS directory. It finishes by telling you that its has completed the back ups. The last "SAY" message may have actually appeared BEFORE the operations which it claimed to have completed if there hadn't been a DELAY statement.

INI_File_Settings

There are various entries in the file WC.INI which determine the way in which WCL works. Below is an explanation of the purpose of each of the settings, including the defaults;

DO=0

With regard to the way WCL runs external programs, some users have complained about the fact that WCL sometimes changes to the directory in which the executable for the program was found, before running it. You can circumvent this feature now. This is implemented when the user types "DO" before the name of the program to be executed. This is explained in more detail under the "DO" command.

You can now avoid having to type "DO" all the time by setting the "DO=" line in WCL.INI to "1", or "ON" (i.e., "DO=1"). With this setting, WCL will behave as if you have typed "DO" before the name of *every* program that you want to execute. This setting is NEW (with version 7.0), and is disabled by default.

Note that turning on this setting may cause some programs not to work properly by virtue of not being able to locate their own data files. Thus it may be better to leave this setting disabled, and to use the "DO" command on a case-by-case basis.

WINDOWLENGTH=50

(This is the length of the WCL window. You can reduce of increase the number from 50. Note that to have enough space for typing commands, 42 is the suggested minimum).

LOCATION-HORIZONTAL=1

(this is the location of the LEFT HAND side of the WCL window. By default, this is the left edge of the screen. You can increase this if you want the window to be moved to the centre, or the right side of the screen for example.

NOTE: assuming that the screen width is 80 characters, for a Standard VGA screen, multiply each character by 8. So, for the left side of the window to be moved to the CENTRE of the screen for example, you can put LOCATION-HORIZONTAL=320).

The EFFECT of this setting depends entirely on the RESOLUTION of your screen. So for SuperVga modes (e.g.800x600; 1024x768) you will need to increase the multiplication ratio.

The easiest way of setting this is to move the WCL window to the desired location, leave it there, and then type "SAVE" at the WCL command prompt. This will cause the location of the WCL window to be saved in WCL.INI.

Note that when you change this setting, you have to allow for the length of the WCL window as set in WINDOWLENGTH (above).

LOCATION-VERTICAL=1

(this is the location of the TOP of the WCL window. By default, this is set to the top of the screen. You can increase it if you wish to move the window DOWN, perhaps to the bottom of the screen.

NOTE: assuming that the screen length is 25 lines, for a standard VGA screen, multiply each line by 19. So, to move the window to the bottom of the screen, you can put LOCATION-VERTICAL=475).

This setting determines the TOP of the WCL window. The window itself occupies about 6 lines. So, you effectively have only 19 lines to play with. In the example given above, 475 is the 25th line of the screen. If you use that setting, you WILL NOT SEE any part of the WCL window (it the rest of it will be below the bottom of the screen). The safe setting for the bottom of the screen on Vga mode (640x480) is LOCATION-VERTICAL=361

For SuperVga modes, you again have to increase the multiplication ratio.)

WCL-PROMPT=PG

(This is the default mode of the WCL command line prompt. It displays the current Drive and Directory (like DOS). If this line is empty, then this is still the default prompt. If you wish to customise the WCL environment, you can change this setting. Anything after the "=" sign is taken LITERALLY and will appear EXACTLY as written. The only exception is "PG" which simulates the ubiquitous DOS prompt.

So you can simulate the famous DBase "dot prompt" by putting on this line, "WCL-PROMPT=."

You can also simulate the UNIX % prompt by "WCL-PROMPT=%" Alternatively, use your own name, "WCL-PROMPT=JOE BLOGGS>" (Note: The longer the prompt, the LESS space you have at the command line for typing commands)

If you want a space to appear after your prompt, add a hash ("#") to the end of the prompt. e.g. "PROMPT FRED>#" -- this will change the prompt to "FRED> ") To return the prompt to one that shows the current directory, type "PROMPT PG". Any prompt that is not PG is taken literally. No other "$" setting is supported. Please note this fact. Note also that you should NOT use the hash for "PG".

You can change the prompt at the command line at any time by using the "PROMPT" command.

e.g. "PROMPT %" or "PROMPT .#"

This change will be saved into WCL.INI if you quit WCL through one of its own exit commands (e.g., "EXIT", "HALT", "QUIT")

DEFAULT-FONT=0

This is the setting for the default font used in the WCL window. When the DEFAULT-FONT is set to ZERO (default) then the System Fixed Font is used. Other possible values are; 1 (this means use the ANSI fixed font); 2 (this means use the OEM fixed font); 3 (this means use the default font for screen device). Any other setting is ignored in favour of the default. Note that the font used in number 3 is often VERY ugly.

WINDOW-BORDER=0

This setting decides whether the WCL main window has a border. If this is set to "1" or "ON", then the window will have a border, and will be resizable. Otherwise, the size of the window will be fixed.

EMULATE-DOSKEY=1

This setting turns on support for a LIMITED emulation of the DOSKEY function of scrolling through a list of past commands (by using the arrow keys - LEFT or UP arrow keys, or CTRL-Z for ("up") and DOWN or RIGHT arrow keys, or CTRL-X (for "down")).

This setting also turns on support for assigning commands to FUNCTION KEYS (F1-F9, and F11-F12); and the CONTROL KEY + AN ALPHABET (CTRL-A to CTRL-Y, but MINUS CTRL-C, CTRL-G, CTRL-H, CTRL-M, CTRL-T, and CTRL-X - these are reserved). F10 also is reserved for use by Windows. You can put assign any command to any of the free keys by just putting the command after the "=" sign. WCL will execute the command EXACTLY as it appears here.

These key assignments are carried out manually in KEYS.WCL, which is just a plain ASCII file that can be edited with the Notepad or any other text editor.

A setting of "1" or "ON" turns this setting ON. Any other setting turns it OFF You can also turn it on/off in WCL by typing "DOSKEY ON" or "DOSKEY OFF"

One limitation is that unless you have enabled the LINE-EDITING functions (see below) you can only edit the commands by using the BACKSPACE key.

LINE-EDITOR=1

This setting turns the option command-line editing ON or OFF. A setting of "1" or "ON" enables the function, and any other setting disables it. You can also turn command-line editing off and on temporarily by using the new command "LINE-EDITOR ON" (to turn it on) or "LINE-EDITOR OFF" (to turn it off).

If DOSKEY emulation is not enabled, command-line editing will not work. When command-line editing is enabled, you can use the left and right arrow keys, Backspace, Delete, Home, and End, to move about and edit the commands being entered, and also to edit the commands that have been brought up through DOSKEY scrolling. Thanx to Stephen Ryan for the line editing function.

When command-line editing is enabled (the default setting in WCL.INI), the text being typed at the command line will scroll horizontally on the same line, and you can move left or right with the cursor keys. If the prompt is less than 20 characters from the end of the WCL window, the cursor will move to the next line to allow more room for the commands to be typed.

When command-line editing is turned off, then the text being typed will wrap onto the next line. In this case, you cannot go back to the previous line.

SHOW-TIMER=0

This setting dictates whether the on-screen clock is displayed or not. See also the "TIMER" command.

TIMER-ON-DESKTOP=0

This setting dictates whether the on-screen clock (if enabled) is displayed on the Windows Desktop or not. See also the "TIMER-ON-DESKTOP" command.

TEXT-COLOR=

This setting determines the color of the text in the WCL windows. You can use any color here.

TEXT-BACKGROUND=

This setting determines the color of the background of the WCL windows. Although you can theoretically use ANY color here, practically, you have to use the same color as you use for your "wallpaper" setting (below).

Notes: for TEXT-COLOR and TEXT-BACKGROUND

These settings should contain whole numbers (i.e., without decimals, or commas) that represent the colors that you want for the text and the window background respectively. Since users may be using any number of display cards and any number of screen drivers at any time, the color codes will vary significantly between systems.

If you ever want to return to the default BLACK text on WHITE background, leave these settings empty, and WCL will use the defaults. Also, because of difficulties that users may face with the use of numbers, I have decided to support a number of non-numerical color codes which are constant on systems with 16 or more colors. The colors that you may specify by NAME for the text color and the text background are;

cyan

white

black

red

green

blue

yellow

magenta

gray

lightgray

darkgray

darkyellow

These can be entered in uppercase or lowercase letters - it does not matter.

If you want to use any color other than the above, then you have to use a numeric value that represents its color. Sorry, I can't help you further here. But if it helps, you can use hexadecimal values (ones that begin with $00, and then are followed by SIX values). C or Pascal programmers will be familiar with these. The six values that follow the $00 are RGB values, but used backwards (i.e., the first two for BLUE, the next two for GREEN, and and the last two for RED). "FF" turns the value to full intensity, and "00" turns the color off. Any number between those two will vary the intensity.

e.g.,

$00000000 = black

$00FFFFFF = white

$00FF0000 = blue

$00808080 = gray

$006F9FFF = lightgray

$000000FF = red

$0000FF00 = green

WALLPAPER=

This setting determines the color of the "brush" that Windows uses to paint the background of the main window. This can take one of 5 colors;

black

gray

white

gray

lightgray

darkgray

NOTES:

1. You should not use numeric values here.

2. You should use the same color here as you used for your text background, otherwise you will get very odd screens in all your WCL windows, and unsightly flashes in the BIGWCL main window.

NOTE:

The three above settings can be set either by manual editing of the WCL.INI file, or by using the new "COLOR" command, which writes the settings to WCL.INI for you. They then take effect when next you run WCL. If you want to see the colors which are already set in WCL.INI, type "GETCOLOR".

The syntax is COLOR <textcolor> <textbackground> <wallpaper>

e.g., "COLOR yellow black black"

SAVE-DESKTOP=0

(this setting *ONLY* has effect when WCL.EXE is your Windows Shell. If set to 0, then the desktop is NOT saved when you exit from WCL. If set to 1 (one) then WCL will save the current state of the Windows Desktop (i.e., all active programs) when you exit. The next time you run Windows, WCL will automatically restore the Windows Desktop to the position it was when you last quit from WCL (i.e., all those active programs will be run automatically). The desktop is saved in a file called "WCL.DSK" in the Windows Directory. This file is in a binary format, so please NEVER try to EDIT it with a text editor. You can of course delete it any time you want.

NOTE: The Desktop will only be saved when you exit from WCL by using one of the WCL exit commands (i.e., "QUIT", "EXIT", "HALT", or pressing the ESCape key). If you exit by pressing Alt F4 or by selecing "Close" from the system menu, then the Desktop will NOT be saved. If you are fond of exiting Windows programs in these ways, then you can save the Desktop manually by typing "SAVE" at the WCL prompt, immediately before quitting.

WINDOWLENGTH.BIG=75

This is the setting for the length (or width) of the BIGWCL.EXE main window. In order for the DIR/W command to work properly, this setting should be at least 75 (note:

that DIR/W is not supported in WCL.EXE).

WINDOWHEIGHT.BIG=200

This is the setting for the height of the BIGWCL.EXE main window. Any setting lower than 25 will be ignored. If you want a scroll back buffer in the BIGWCL main window, this setting can be increased - the higher the setting, the bigger the scroll back buffer (e.g., a setting of 250 will give a scroll back buffer of about 7 screens).

This setting should not be higher than 250. In fact, any setting higher than 250 will be reduced to 250 by BIGWCL. I have done this because some many users experience problems with strange screen behaviour with higher settings.

LOCATION-HORIZONTAL.BIG=1

This is the location of the TOP LEFT corner of the BIGWCL window.

LOCATION-VERTICAL.BIG=1

This is the location of the TOP of the BIGWCL window.

BIGWCL-DEFAULT-FONT=0

This is the setting for the default font used in the BIGWCL window. When the DEFAULT-FONT is set to ZERO (default) then the System Fixed Font is used. Other possible values are; 1 (this means use the ANSI fixed font); 2 (this means use the OEM fixed font); 3 (this means use the default font for screen device). Any other setting is ignored in favour of the default. Note that the font used in number 3 is often VERY ugly.

STARTUP1= ; STARTUP2= ; STARTUP3= ; STARTUP4=

(These are for indicating the programs to be loaded by WCL every time you start a WCL session. They are looked at *ONLY* if WCL is your Windows shell. If you retain Program Manager as your Windows shell, these lines are COMPLETELY IGNORED. WCL does not load the programs in your Windows start up group file, and this is the way of compensating.

So if for example, you want CONTROL.EXE to be loaded every time you start WCL, you can put "STARTUP1=CONTROL.EXE". If the programs are not in the DOS path, then you have to type in the FULL PATH of the program (e.g. "STARTUP2=D:\MYDIR\MYPROG\MYPROG.EXE"). NOTE that the program name/path must not exceed 78 characters, or it will be truncated.

Only 4 start up programs are supported here. If you must have more than 4, then put one on the line that reads "RUN=" and one on the line that reads "LOAD=" in your WIN.INI file. Alternatively you can create a WCL batch file (e.g., "STARTUP.CBF") and include its name in one of the startup lines. With such a batch file, you can load as many programs as you wish.

NOTE: The programs contained on the start up lines will be loaded EVERY TIME that a WCL session is commenced. So, if for any reason you are running multiple copies of WCL, EACH copy will load all the programs. So it is better to leave those lines as they are until you have finished configuring WCL for your system. If you are likely to want to run multiple copies of WCL, do not add anything to these lines).

CONFIRM-OVERWRITES=0

This sets the behaviour of the File Copy routines. When set to 0 (zero; this is the default) existing files will be overwritten by the versions being copied, WITHOUT WARNING (this is like the DOS Copy command).

If set to 1 (one) then you will ALWAYS be prompted for confirmation before an existing file is overwritten.

NOTE: The WCL.INI file is polled for this setting only ONCE (when the program is loaded) so any change you make to this setting will take effect only after you run WCL again (or if you run another copy of it by typing "WCL" at the prompt, and Close the original copy)

BACK-UP-INI-FILES=1

This setting is effective *only* if WCL is your Windows Shell. If set to 1, then WCL will back up your WIN.INI and SYSTEM.INI files everytime Windows is started. WCL tries to make TWO backups of each file, with extensions of ".WCL". This is useful for restoring your Windows setup in cases when some program has messed up your INI files. If set to 0 (zero) then the backups will not take place.

IMPORTANT NOTES

[1] There must be NO SPACES AT ALL between the entries on each line. e.g "WINDOWLENGTH = 50" is NOT valid because there are spaces both before and after the "="

[2] The responsiblity for suppying correct and sensible values for these window co-ordinates is TOTALLY YOURS. The default settings are quite adequate for most needs, and you can always re-locate and re-size the window using your mouse.

[3] If WCL cannot find the file WCL.INI at startup time, then the default values explained above will always apply.

[4] The WCL.INI file is only read ONCE - when WCL is loaded. If you change any thing in the file, you will have to close and restart WCL for the changes to take effect. The only exception to this is with respect to directory aliases. When you use the "GOTO" command, WCL searches always WCL.INI for an alias for the name you supply.

[5] If you are experimenting with different settings for the WCL window, there is nothing to stop you from testing your settings on another copy of WCL. You can run another copy of WCL by typing "WCL" at the prompt. You will then see another copy loaded, and reflecting the window co-ordinates in the current version of WCL.INI. NOTE that if you have not changed the window coordinates, then the other copies of WCL will have their windows right on top of the current copies.

[6] If WCL is NOT your Windows shell, then the Start Up lines and SAVE-DESKTOP setting in WCL.INI will be ignored.

Disclaimer

The WCL programs are supplied AS IS, without ANY WARRANTIES WHATSOEVER. I will accept NO RESPONSIBILITY for any loss or damage, financial or otherwise, consequent upon the use or purported use of WCL for any purpose whatsoever.

If these terms are NOT acceptable to you, then you have no licence to use or test WCL. You should DELETE the programs from your disks immediately.

Registration

WCL is distributed under the Shareware principle. The shareware version can be copied and distributed freely, as long as ALL the supplied files, including documentation (this file) are included, and NO ATTEMPT is made to modify any of the files. The program may not be supplied or bundled with any COMMERCIAL application without prior WRITTEN permission from me.

The Shareware principle means that you get a chance to EVALUATE the program free of charge for a reasonable period of time (usually 30 days). It does not mean that you will NOT have to pay for the program. If you find WCL useful and would like to continue using it then you are requested to please REGISTER your copy with the author.

REGISTRATION FEE:

£18.00 (U.K. STERLING)

$26.00 (U.S.)

$30.00 (Canadian): British Columbia residents should add 7% Sales Tax

$40.00 (Australian)

Kr170.00 (Danish)

Prices are subject to change without notice.

If you are outside the U.K. and Ireland, please look in the section titled "REGISTRATION SITES" (below) for the details of registration sites which are close to you, and in which you can register in your own currency, or by credit card. This will save you having to send foreign cheques to me, thereby incurring bank charges.

Registration_Sites

Users from outside the United Kingdom who wish to send their registration fee to me in the U.K. should please send or an International Money Order drawn out in STERLING only. Otherwise, if sending cheques drawn out in currencies other than Sterling, please add £3 (Three Pounds Sterling) to cover U.K. bank charges. Thus, for example, if you are sending a cheque that is drawn on a bank in the U.S.A, you need add $4.50 to the registration fee to cover U.K. bank charges (meaning $US30.50).

Please note that the extra £3 is COMPULSORY if you are sending me a cheque that is not drawn out in Sterling.

NOTE:

If you specify that the transmission of your registered copy should be by UUENCODED E-MAIL, there is NO CHARGE for handling and carriage. If you want your registered copy to be sent to you on a diskette, please ADD £3 (Three Pounds Sterling, or equivalent (e.g., $US4.50) to cover handling and carriage.

REGISTRATION will entitle you a copy of the most current version, minus the Registration messages, and a 50% discount on upgrades.

Please note that there is no printed manual.

On registration, you will also get;

1. A copy of the most recent update of the program, licensed to you, and without all the registration messages.

2. Support for the program

3. A clear conscience

4. The satisfaction of being an honest person

5. Complimentary copies of some of my Windows utilities, namely;

[a] EXITWIN.EXE (click on the icon of this program, and you quit Windows - no fuss, no warnings, no questions)

[b] RUNWIN.EXE (click on the icon of this program, and Windows is shut down and restarted again - useful for those occasions when you need to reboot Windows, e.g. you have just installed a new device driver, or something has corrupted Windows, and you want to restart it).

If you wish to Register your copy, please send the payment to me at the address below, or to one of my registration sites;

Dr. A. Olowofoyeku,

268 Horwood,

Newcastle,

Staffordshire, ST5 5BQ

ENGLAND.

Internet:

laa12@keele.ac.uk

chief@mep.com

Please specify floppy disk size (3.5" or 5.25").

Registration_Sites
Registration_FORM

Registration Sites

NOTE:

If you specify that the transmission of your registered copy should be by UUENCODED E-MAIL, there is NO CHARGE for handling and carriage. If you want your registered copy to be sent to you on a diskette, please ADD £3 (Three Pounds Sterling, or the equivalent in the currency in which you are paying - e.g., $US4.50) to cover handling and carriage.

Registration_FORM

YOU CAN SEND THE REGISTRATION FEE TO ANY OF THE FOLLOWING REGISTRATION SITES;

CANADA, AND NORTH AMERICA

Minds Edge Productions Inc.
P. O. Box 211
3456 Dunbar Street
Vancouver, BC V6S 2C2
Canada

Internet: info@mep.com

Fidonet: 1:153/769

BBS: (604) 261-6144

Fee: $26.00 (US funds)

or: $30.00 (Canadian funds)

plus: $4.50 (for handling - not applicable to transmission by UUencoded E-mail)

Method of payment: Checks, Money Orders

Make cheques/money orders payable to: "Minds Edge Productions Inc.".

British Columbia residents should add 7% sales Tax.

UNITED STATES

TED KRAUS
Synergy Online Communications Inc.
P. O. Box 2630
Mercerville
NJ 08690
U.S.A

Tel: (609) 587 6200

1-800-732 5856 (within the U.S.A only)

BBS: (908) 730 9002

Internet: ted.kraus@syncomm.com

Fee: $26.00 (US funds)

plus: $4.50 (for handling - not applicable to transmission by UUencoded E-mail)

Method of payment: Checks, Money Orders, Visa, Mastercard, American Express.

Make cheques/money orders payable to: "TKO/ Real Estate Advisory Group Inc."

UNITED STATES

TODD MERRIMAN
Software Toolz, Inc.
8030 Pooles Mill Dr.
Ball Ground,
GA 30107
U.S.A.

Fax: 404-887-5960

Internet: software@toolz.atl.ga.us

Fee: $26.00 (US funds)

plus: $4.50 (for handling - not applicable to transmission by UUencoded E-mail)

Method of payment: Checks, Money Orders, Visa, Mastercard, American Express.

SCANDINAVIA AND NORTHERN EUROPE:

HENRIK MOERK
Survival BBS
P.O.Box 1538
DK-2700 Bronshoj
Denmark

FIDO: 2:231/306

Internet: Lene@vax.psl.ku.dk
Hmk@research.novo.dk

Fee: Kr170.00 (Danish funds)

plus: Kr25.00 (for handling - not applicable to transmission by UUencoded E-mail)

Method of payment: Cheques, Eurocheques, Money Orders,

GIRO: 1-207-4247

Make cheques/money orders payable to: "HENRIK MOERK".

AUSTRALIA, NEW ZEALAND, ASIA, AND THE FAR EAST

DAVID PERKOVIC
DP Computing
P.O.Box 712
Noarlunga Center

SA 5168
Australia

Internet: perkovic@cleese.apana.org.au
dpc@mep.com
Tel: +61 8 326 4364

Fee: $40.00 (Australian funds)
plus: $6.00 (for handling - not applicable to transmission by UUencoded E-mail)

Method of payment: Cheques, Money Orders
Make cheques/money orders payable to: "DP Computing".

UNITED KINGDOM, IRELAND, EUROPE, AND EVERYWHERE ELSE

Dr. A.A. OLOWOFOYEKU
268 Horwood,
Newcastle,
Staffs, ST5 5BQ,
ENGLAND.

Internet: laa12@keele.ac.uk
chief@mep.com
Fee: £18.00 (U.K. funds; or equivalent)
plus: £3.00 (only if sending a foreign cheque)
plus: £3.00 (for handling - not applicable to transmission by UUencoded E-mail)

Method of payment: Cheques, Eurocheques, Money Orders,

To register WCL, please PRINT and FILL IN the following Registration FORM

To:

_____

_____

_____

_____

I wish to REGISTER my copy of "WCL"

I would like my registered copy sent to me by;

[a] 5.25"[ ] 3.5"[ ] Floppy Disk. (please tick)

OR

[b] by UUencoded e-mail [ ] (Internet addresses only)

(please note that the UUencoded ZIP file is about 400kb in size)

I use the "big" [ ] "small"[ ] version (please tick)

I am paying the REGISTRATION FEE of _____

ADD Tax (if applicable) _____ (see the info on registration sites)

ADD £3 for handling and carriage _____ (NO CHARGE for UUencoded E-mail)

Total FEE: _____

I am paying by Cheque/Money Order/Credit Card (delete as inapproriate)

NAME _____

ADDRESS _____

_____

_____

POST/ZIP CODE _____

E-MAIL _____

How did you get your copy of WCL?

_____

IF PAYING BY CREDIT CARD, PLEASE SEND THE FOLLOWING DETAILS;

(NOTE: Not all sites accept credit cards so please refer to the list of REGISTRATION SITES)

CARD ISSUER _____

CARD NUMBER _____

DATE OF ISSUE _____

EXPIRY DATE _____

SIGNATURE _____

DATE _____

PAUSE

PAUSE - For use in WCL .CBF batch files. See the "Batch Files" section.

Batch_Files

DELAY

DELAY - For use in WCL .CBF batch files. See the "Batch Files" section.

Batch_Files

EXECWAIT

EXECWAIT - For use in WCL .CBF batch files. See the "Batch Files" section.

Batch_Files

DOS PROGRAMS

DOS programs can be run from WCL as easily as Windows programs. You can either create a PIF file for the DOS program, with the PIF Editor, or run the program's binaries directly. NOTE that when running a DOS program for which there is no specific PIF file, Windows will use the settings in a file called _DEFAULT.PIF (note the underscore) which exists in the Windows directory. You can either change the settings in this PIF file to achieve a standardised setting for your DOS programs, or you can delete the file. If you choose the latter option, then Windows will use some other defaults for running your DOS programs.

If you ever have any problem running some DOS programs under WCL, check first for the settings in the _DEFAULT.PIF file, because this is usually where the problem lies. See also the SPAWN, DO, RUNDOS, and TIMER commands (below).

NOTE: there are problems if you try to run FULL SCREEN DOS programs when the timer feature is ON - often you just get dumped unceremoniously to the DOS command prompt. Thus, if you are going to run FULL SCREEN DOS programs from WCL, then you should NOT enable the timer feature, or you should type "TIMER OFF" before running the program.

DO
RUNDOS
SPAWN
TIMER