

Windows Command Help Index

Basic Information

- [Introduction](#)
- [Batch Files](#)
- [Command Execution](#)
- [Command Options](#)
- [Command Recall](#)
- [Executing Windows Command](#)
- [Filename Completion](#)
- [Multiple Commands](#)
- [Processing the Command Line](#)
- [Redirection](#)
- [Using Help](#)
- [Using the Keyboard](#)
- [Using the Menu](#)
- [Wildcards](#)

Commands

Alphabetical

- [Command Index](#)

Categorical

- [Batch Commands](#)
- [DOS Commands](#)
- [File Commands](#)
- [Windows Commands](#)
- [*WindowsCommand* Commands](#)

Reference

- [Error Messages](#)

Introduction

Windows Command is a command-line interface designed for Microsoft Windows. Just as Windows is not a replacement for all DOS-based programs, Windows Command is not meant to be a replacement for COMMAND.COM. It is a method of interfacing with Windows in a familiar command-line environment. It can be used to minimize the use of DOS-based command lines while in Windows, and can also be used as a replacement for shells like Program Manager or File Manager.

Windows Command is a blend of a command line and a graphical program. It features pull-down menus with a submenu of all the Windows Command commands listed to "pull down" onto the command line, a submenu of all programs shipped with Microsoft Windows, so that they may be executed with a menu selection, and a submenu for access to the Windows Command help facility. Windows Command also features a pop-up command history dialog that allows point-and-shoot access to previously executed commands.

Windows Command is a powerful command-line interface. It has many features which make using the command line easier, such as multiple commands per line, command recall, filename completion, and command aliases. It also has powerful command processing functions, such as redirection of standard output, and wildcard support.

Windows Command has been designed to allow keyboard access to all features of its own, and also many features of Windows, without the use of a mouse. It allows executing, minimizing, maximizing, and closing Windows programs from the command line, as well as a number of other Windows functions, such as arranging icons on the desktop, and managing file associations, all using simple commands, and without touching a mouse.

This help file assumes that you have basic knowledge of DOS and of Windows. If you are new to this system, you may want to keep your DOS and Windows manual handy for reference.

Windows Command is a user-supported program. If you have any suggestions, comments or problems concerning Windows Command, I invite and request you to write to me at the address below:

Michael B. Tierney
205 Inglewood Dr.
Pittsburgh, PA 15228

Thank you for using Windows Command. I hope you find it to be a useful tool, and a productive new way to use the Microsoft Windows environment.

Using the Menu

The Windows Command menu has three main submenus: Commands, Execute, and Help. Each of these allows access to powerful abilities within the Windows Command environment.

Commands

This menu selection contains categories of listings of all the available commands, an **Exit Windows** selection, to exit the Microsoft Windows environment, and an **Exit** selection, to quit Windows Command. The command listing is separated into categories: **Batch** file commands, **File** system commands, **DOS** function commands, **Windows** environment commands, and **Windows Command** environment commands.

Selecting one of the commands from the menu will bring that command down to the command line. Thus, selecting "Dir" from the "File" category will make "Dir" appear on the command line. At that point, you can press F1 to bring up the help screen for that command, or hit <ENTER>, and the command will execute.

Execute

This menu selection contains a listing of all the Windows programs shipped with Microsoft Windows, and allows you to execute any given program by selecting it from the list.

Help

This menu selection allows access to the Windows Command help system. It includes an **Index** selection, which will bring up the Windows Command Help Index, a **Basics** selection which will bring up the Introduction to Windows Command, a **Keyboard** selection, which will bring up the available command keys, **Using Help**, which will show how to use Windows Command Help, and **About**, which shows author information for Windows Command.

Batch Files

A batch file is a text file which contains a listing of commands to be executed in order. You can create such a file using Notepad, which comes with Windows. Commands are entered one-per-line in the file, with a maximum command length of 255 characters. "Windows Command Batch" files must be given the file extension ".WCB".

When you first start Windows Command, it searches for and executes the file AUTOEXEC.WCB, if it is found in any directory on your DOS path.

Windows Command has several commands available specifically for batch execution. The **if** command allows conditional execution of commands, and **Goto** will branch execution to a *label*.

A batch file label consists of a colon ":", followed by the label name. For example, **:ThisLabel** is a label named ThisLabel, and the command **Goto ThisLabel** will subsequently start batch file execution at the label ThisLabel.

A sample batch file follows:

```
@echo off
echo This is a test batch file.
goto MakeTheTestFile
:EraseTheTestFile
echo Erasing the test file...
erase testfile.$$$
if not exist testfile.$$$ goto EndTheTestBatch
:MakeTheTestFile
echo Making the test file...
dir > testfile.$$$
goto EraseTheTestFile
:EndTheTestBatch
echo Ending the test batch file...
exit
```

Execution begins with **@echo off**, which will suppress the batch file commands from appearing on the screen. Next, "**This is a test batch file...**" will be echoed to the screen, then execution will go to the label **MakeTheTestFile**. "**Making the test file...**" will echo to the screen, and then the output from a **dir** command will be routed into the file **TESTFILE.\$\$\$**. Execution will branch to **EraseTheTestFile**, "**Erasing the test file...**" will echo to the screen, then **TESTFILE.\$\$\$** will be erased. At that point, the file **TESTFILE.\$\$\$** will cease to exist, so that when **if not exist testfile.\$\$\$ goto EndTheBatchFile** executes, the batch file will branch to EndTheBatchFile

Important Note:

Do not write batch files that either call themselves infinitely or use an infinite GOTO loop. This can make the batch file execution dominate your system resources and crash or halt your system. If you have problems such as this with batch files, the cause is most likely an infinite loop.

Wildcards

Wildcards provide a way to run a command on numerous files without having to type in each file name. Commands which support wildcards are: Attrib, Copy, Dir, Erase, Except, Move, Print, Rename, Rmdir, and Type.

The wildcard characters are * and ?. For example, "*.DOC", references all files in the given directory that have the extension ".DOC". "??.DOC" references all files in the given directory that have the extension ".DOC", and a filename that is two characters or less long. "MYFIL?.*" references any file whose filename begins with "MYFIL", and is no more than 6 characters long, and has any file extension. "*" is any file with any filename and no extension, and " *.*" is any file with any filename and extension.

Note: "*ANY.*" will be read as " *.*", since any characters following an asterisk in either the filename or extension are ignored.

Processing the Command Line

The command line is first checked for the multiple command character, the caret (^). If you need to use a caret for any other reason, put the whole expression inside single or double quotes. For example, to delete the file THIS^FIL.DOC, you will need to enter ERASE "THIS^FIL.DOC" or ERASE 'THIS^FIL.DOC'. Otherwise, the command will be split up and executed as ERASE THIS, and then FIL.DOC.

Next, the command line is checked for redirection, and the redirection is performed. The standard output redirection symbols are ">" to begin a new file, or ">>" to append the file.

The command is checked for a leading at sign (@), which means that the command will not be echoed to the screen, even if echo is on.

Finally, the command line is split into arguments. "White space" characters divide arguments, as well as command option delimiters. "White space" characters include semicolons (;), commas (,), spaces, and tabs. Command option delimiters include the foreslash (/), and the dash (-), since any command option in Windows Command can be set apart by either a foreslash or a dash.

The command is then executed.

Command Execution

Windows Command tries several methods of finding the executable program when processing the command line:

File Extension Completion:

If a file extension is not given when you type the program name, Windows Command will try to find the first executable on your DOS path or given path with the matching extensions .WCB, .BAT, .PIF, .EXE, and .COM, searching for files with extensions in that order. This means that if you type XYZ, and both XYZ.WCB and XYZ.COM exist, XYZ.WCB would execute. Likewise, if only XYZ.EXE and XYZ.COM existed, XYZ.EXE would execute, etc.

Command Extension Matching:

If a file extension is given when you type the program name, the extension is first checked to make sure it is an executable (with extensions .WCB, .BAT, .PIF, .EXE, .COM), and if so, the file is executed. Otherwise, the extension is checked for a file association. If an association is found, the associated program is started with the given file as the startup file. If no association is found for a non-executable file, the filename is returned as an Unknown Command.

Redirection

Windows Command supports the redirection of standard output. This means that any text from any command that appears in the Windows Command window can be routed into a file. The redirection symbols are ">", which erases the file if it exists and then puts the new text into it, and ">>", which appends the new text to the file, whether it exists or not. The redirection symbol goes at the end of the command, and is followed by the file name that you want the output to go to.

For example, **DIR > DIRFILE.TXT** will create a file called DIRFILE.TXT that contains the output of the Dir command.

Multiple Commands

Windows Command allows more than one command to be executed on a line. The separational character, the caret (^), marks the beginning of a new command on a line. For example, **DIR ^ ECHO HI THERE** will execute as if you entered **DIR**, and then **ECHO HI THERE**.

When you use multiple commands, if a command fails, the remainder of the line does not execute. For example, if you entered **DIR /X ^ ECHO HI THERE**, **DIR /X** would exit with an error (Invalid Parameter), and **ECHO HI THERE** would never execute. This allows you to structure a series of commands that require successful completion to continue.

Command Options

Many of the commands have options which modify the way they execute. These are all listed in the individual commands' help screens. The help screens use the convention of the foreslash (/) to show command options, however, all commands will accept either a foreslash or a dash (-) as a command option delimiter. For example, **DIR /W** and **DIR -W** will execute exactly the same way.

Windows Command Execution

You can specify a command to execute on startup by typing the command as a parameter when you run Windows Command. For example, typing WCOMMAND DIR /W ^ CLS in the Run dialog of Program Manager or File Manager (or whatever you use to start Windows Command) will cause Windows Command to start, and execute DIR /W, and then CLS.

Normally, when you start Windows Command, it searches for and executes the AUTOEXEC.WCB batch file if it exists. However, if you specify a command on the command line (which could be another batch file), AUTOEXEC.WCB will not be executed on startup, and only the given command will be run.

On first execution, Windows Command detects the size of your display and automatically selects the font (Courier or System) which will be most readable.

Filename Completion

With filename completion, you can search for a filename and insert it into a command by scrolling through file names. First, you enter none or as much of the filename as you want to search for, then press **F9** or **TAB**. The filename will be inserted automatically into the command line, and you can continue scrolling through different filenames pressing **F9**. If you go too far, you can scroll backwards through the files by pressing **F8**.

For example, if you want to open Notepad to edit the file SYSTEM.INI in your Windows directory:

- if you are in another directory, type **NOTEPAD \WINDOWS**, make sure the cursor is on the partial filename, or on the space to the right of the last character, and hit **F9**. Continue pressing **F9** (or press **F8** if you accidentally go too far) until you reach WIN.INI. Then press **ENTER** to process the command.
- if you are in your Windows directory, type **NOTEPAD S** or just **NOTEPAD** (with a space after the last letter), make sure the cursor is on the partial filename, or on the space to the right of the last character, (or to the right of the space after "NOTEPAD") and hit **F9**. Continue pressing **F9** (or press **F8** if you accidentally go too far) until you reach WIN.INI. Then press **ENTER** to process the command.

Command Recall

Command Recall allows you to recall previously entered commands to edit or execute them again. Pressing the **UP** arrow key will recall the last command entered. Pressing **UP** repeated times will continue back to the first command, and then wrap around to the last command again. Once **UP** has been pressed, pressing **DOWN** will recall the commands in reverse order. You can view the entire command history list by entering the History command.

Command Editing Keys

The following are special keys used by Windows Command:

Left	Move cursor one character to the left
Right	Move cursor one character to the right
Home	Move cursor to beginning of command line
End	Move cursor to end of command line
PageUp	Scroll display up one page
PageDown	Scroll display down one page
Enter	Enters the command to process
Insert	Toggles insert mode
Delete	Deletes character highlighted by cursor
Backspace	Deletes character immediately before cursor
Escape	Erases entire command line
Up	Command Recall - Forward
Down	Command Recall - Backward
Tab	Filename Completion - Forward
F9	Filename Completion - Backward
ALT	Accesses the Menu
F1	Help - Context-sensitive help if command line is not empty
Ctrl-Break	Interrupts command activity and resets Windows Command

Using Help

There are a number of ways to access the Windows Command Help Facility: pressing F1, selecting Help options from the menu, or issuing the Help command at the command line.

When you press F1, Windows Command looks at the current command line. If the line is empty, the Help Index is displayed. If there is a command on the command line, the first word is read, and the help screen for the appropriate command will be displayed.

The Help submenu gives instant access to the Help Index, the Introduction screen, the Using the Keyboard screen, and this screen, Using Help.

The Help command issued at the command line can display either the Help Index or context-sensitive help.

Batch Commands

These commands are used in Windows Command Batch files:

<u>Goto</u>	Branch to a Label
<u>If</u>	Conditional command Execution
<u>Pause</u>	Pause Until User Presses a Key
<u>Rem</u>	Remark or Null Command

File Commands

These commands control files and directory operations:

<u>Attrib</u>	Get/Set File Attributes
<u>Cd</u>	Change Current Directory
<u>Chdir</u>	Change Current Directory
<u>Copy</u>	Copy a File
<u>Del</u>	Erase a File
<u>Dir</u>	Display a Directory of Files
<u>Erase</u>	Erase a File
<u>Except</u>	Execute on All Except Specified Files
<u>Md</u>	Make a New Directory
<u>Mkdir</u>	Make a New Directory
<u>Move</u>	Move a File
<u>Rd</u>	Remove a Directory
<u>Ren</u>	Rename a File
<u>Rename</u>	Rename a File
<u>Rmdir</u>	Remove a Directory
<u>Type</u>	Display a file on the screen
<u>Vol</u>	Get a Disk Volume Label

DOS Commands

These Commands give information on, or work through the Operating System:

<u>Date</u>	Get/Set the System Date
<u>Mem</u>	Display Free Memory and System Statistics
<u>Path</u>	Display DOS Path
<u>Print</u>	Print a File
<u>Set</u>	Display DOS Environment Variables
<u>Time</u>	Get/Set System Time
<u>Ver</u>	Display System Program Versions
<u>Verify</u>	Get/Set DOS Verify Flag

Windows Command Commands

These commands control the *Windows Command* environment:

<u>?</u>	Access Windows Command Help
<u>About</u>	Shows Windows Command author information
<u>Alias</u>	Set an Alias for a Command
<u>Beep</u>	Beep the Speaker
<u>Break</u>	Get/Set Break Checking Status
<u>Cls</u>	Clear the Screen
<u>Color</u>	Change Screen Colors
<u>Echo</u>	Echo Line, or Get/Set Echo Status
<u>Eset</u>	Get/Set Windows Command Environment Variables
<u>Exit</u>	Exit Windows Command
<u>Help</u>	Access Windows Command Help
<u>History</u>	View and Recall Command History
<u>Prompt</u>	Set Command Prompt
<u>Unalias</u>	Remove an Alias Name

Windows Commands

These commands control the Windows environment:

<u>&</u>	Load and Minimize a program
<u>+</u>	Load and Maximize a program
<u>\</u>	Change Directory and Execute
<u>Arrange</u>	Arranges the icons on the desktop
<u>Assoc</u>	Get/Set file extension associations
<u>Cdx</u>	Change Directory and Execute
<u>Cl</u>	Load a file to the Clipboard
<u>Clipload</u>	Load a file to the Clipboard
<u>Clo</u>	Close a Program
<u>Close</u>	Close a Program
<u>Max</u>	Maximize a Program
<u>Maximize</u>	Maximize a Program
<u>Min</u>	Minimize a Program
<u>Minimize</u>	Minimize a Program
<u>Res</u>	Restore a Minimized Program
<u>Restore</u>	Restore a Minimized Program
<u>Runmax</u>	Load and Maximize a program
<u>Runmin</u>	Load and Minimize a program
<u>Task</u>	Display all Running Programs
<u>Winexit</u>	Exit Windows

Windows Command Command Index

This is a complete list of *Windows Command* Commands.

<u>&</u>	Load and Minimize a program
<u>+</u>	Load and Maximize a program
<u>?</u>	Access Windows Command Help
<u>\</u>	Change Directory and Execute
<u>About</u>	Shows Windows Command author information
<u>Alias</u>	Set an Alias for a Command
<u>Arrange</u>	Arranges the icons on the desktop
<u>Assoc</u>	Get/Set file extension associations
<u>Attrib</u>	Get/Set File Attributes
<u>Beep</u>	Beep the Speaker
<u>Break</u>	Get/Set Break Checking Status
<u>Cd</u>	Change Current Directory
<u>Cdx</u>	Change Directory and Execute
<u>Chdir</u>	Change Current Directory
<u>Cl</u>	Load a file to the Clipboard
<u>Clipload</u>	Load a file to the Clipboard
<u>Clo</u>	Close a Program
<u>Close</u>	Close a Program
<u>Cls</u>	Clear the Screen
<u>Color</u>	Change Screen Colors
<u>Copy</u>	Copy a File
<u>Date</u>	Get/Set the System Date
<u>Del</u>	Erase a File
<u>Dir</u>	Display a Directory of Files
<u>Echo</u>	Echo Line, or Get/Set Echo Status
<u>Erase</u>	Erase a File
<u>Eset</u>	Get/Set Windows Command Environment Variables
<u>Except</u>	Execute on All Except Specified Files
<u>Exit</u>	Exit Windows Command
<u>Goto</u>	Branch to a Label
<u>Help</u>	Access Windows Command Help
<u>History</u>	View and Recall Command History
<u>If</u>	Conditional command Execution
<u>Max</u>	Maximize a Program
<u>Maximize</u>	Maximize a Program
<u>Md</u>	Make a New Directory
<u>Mem</u>	Display Free Memory and System Statistics
<u>Min</u>	Minimize a Program
<u>Minimize</u>	Minimize a Program
<u>Mkdir</u>	Make a New Directory
<u>Move</u>	Move a File
<u>Path</u>	Display DOS Path
<u>Pause</u>	Pause Until User Presses a Key
<u>Print</u>	Print a File

<u>Prompt</u>	Set Command Prompt
<u>Rd</u>	Remove a Directory
<u>Rem</u>	Remark or Null Command
<u>Ren</u>	Rename a File
<u>Rename</u>	Rename a File
<u>Res</u>	Restore a Minimized Program
<u>Restore</u>	Restore a Minimized Program
<u>Rmdir</u>	Remove a Directory
<u>Runmax</u>	Load and Maximize a program
<u>Runmin</u>	Load and Minimize a program
<u>Set</u>	Display DOS Environment Variables
<u>Task</u>	Display all Running Programs
<u>Time</u>	Get/Set System Time
<u>Type</u>	Display a file on the screen
<u>Unalias</u>	Remove an Alias Name
<u>Ver</u>	Display System Program Versions
<u>Verify</u>	Get/Set DOS Verify Flag
<u>Vol</u>	Get a Disk Volume Label
<u>Winexit</u>	Exit Windows

About

Purpose:

Shows the author information for Windows Command.

Usage:

ABOUT

Example:

ABOUT will display a dialog box containing the author information for Windows Command.

Alias

Purpose:

Sets an alias for a command or series of commands. The alias acts as a Windows Command internal command, and will override existing commands of the same name.

Usage:

ALIAS aliasname command(s)

Example:

ALIAS MV MOVE will allow you to type MV to execute a MOVE command.

ALIAS DIR DIR /W will make DIR give a default wide display.

Arrange

Purpose:

Arranges the icons on the desktop.

Usage:

ARRANGE

Example:

ARRANGE will arrange the icons on the desktop so that they are all in neat rows.

Assoc

Purpose:

Get or set associations between a filename extension and the program that uses the file with that given extension.

Usage:

ASSOC <LIST|extension> <program name>

<extension> is the 3-characters following the "." in the filename.

<program name> is the full pathname of the program that uses files with the extension given.

Example:

ASSOC LIST will display a listing of all the existing filename associations.

ASSOC BMP will show the current setting for the BMP extension and prompt you for whether you want to remove that association or not.

ASSOC WCB C:\WINDOWS\WCOMMAND.EXE will create an association between files with the extension ".WCB", and Windows Command. After executing this command, whenever a file with the extension ".WCB" is run from Program Manager, File Manager, Windows Command, or any other program that uses associations, Windows Command will automatically be started.

Attrib

Purpose:

To Display or set the Read-Only, System, Hidden or Archive attributes of a file or subdirectory.

Usage:

ATTRIB [+r|-r] [+s|-s] [+h|-h] [+a|-a] filename [/D]

+r/-r Add/Remove Read Only Attribute

+s/-s Add/Remove System File Attribute

+h/-h Add/Remove Hidden Attribute

+a/-a Add/Remove Archive Attribute

/D Set or Display Subdirectory Attributes

Example:

ATTRIB *.* will display the attributes of all files in the current directory.

ATTRIB -R +S -H +A THISFILE.DOC will add System and Archive attributes to and remove Read-Only and Hidden attributes from the file THISFILE.DOC.

Beep

Purpose:

To Beep the Speaker.

Usage:

BEEP

Example:

BEEP will cause the speaker to sound a tone.

Break

Purpose:

To Display Ctrl-Break Checking Status, or turn it on or off. The default is ON.

Usage:

BREAK [ON|OFF]

Example:

BREAK will show whether break checking is on or off.
BREAK ON will set break checking on.

Cdx or \

Purpose:

To automatically change to the directory in which a program resides when executing a program. This will allow programs which expect to be executed from the same directory as they are in to run properly.

Usage:

CDX <program>
or \ <program>

Example:

CDX C:\123\LOTUS will execute the command LOTUS from the directory C:\123.
\ PROGMAN will start Program Manager from your windows directory.

Chdir or Cd

Purpose:

To change the current directory on the default or specified drive.

Usage:

CHDIR <directory>
or **CD <directory>**

Example:

CHDIR \ will change the current directory to the root of the default drive.
CD A:\ will change the current directory of the A drive to the root directory;
the current drive will remain the same.

Clipload or CI

Purpose:

To load a file into the clipboard. The file size must be less than 64K.

Usage:

CLIPLOAD <filename>
or **CL <filename>**

Example:

CL THISFILE.DOC will load the file THISFILE.DOC to the clipboard.

Close or Clo

Purpose:

To Close a currently active task. A message is sent to the program, telling it to close; whether or not the task actually ends will depend on the application itself. Task IDs can be found by issuing the TASK command.

Usage:

CLOSE <taskid>
or **CLO <taskid>**

Example:

CLO 9337 will close the task with the Task ID 9337.

Cls

Purpose:

Clears the *Windows Command Window*.

Usage:

CLS

Example:

CLS will clear the screen.

Color

Purpose:

To change the colors of the Windows Command Window.

Usage:

COLOR <color> [ON <color>]

<color> is one of the following:

RED
GREEN
BLUE
YELLOW
MAGENTA
CYAN
BLACK
WHITE

Example:

COLOR BLACK ON WHITE will set the text color to black and the background color to white.

Copy

Purpose:

To Copy or Append a file to another file.

Usage:

COPY [/A/B]<fromfile> [+ [/A/B] <fromfile2> ...] <tofile>

/A Copy remaining files in Ascii mode

/B Copy remaining files in Binary mode

Example:

COPY THISFILE.DOC /A + THISTOO.DOC A:\THATFILE.DOC will copy THISFILE.DOC and append THISTOO.DOC to the file A:\THATFILE.DOC.

Date

Purpose:

Set the System Date.

Usage:

DATE <month> <day> <year>

<month> is an integer between 1 and 12

<day> is an integer between 1 and 31

<year> is the full number of the year (i.e. 1960, not 60)

Example:

DATE 7 4 1991 will set the date to July 4th, 1991.

Dir

Purpose:

To display a directory listing of files.

Usage:

DIR [/1/2/4/W/P/B/U/N/T/S/Onedsgu/Adrsha] [<filename>]

- /1 display one column of files (default)
- /2 display two columns of files
- /4 display four columns of files
- /W display five columns of files
- /P pause after each page of files
- /B display only filenames, no other information
- /U display filenames in upper case
- /N do not display file attributes in listing
- /T display only directory size totals and number of files
- /S display all files in current directory and subdirectories
- /O sort order
 - Reverse any of the below orders
 - N by Name
 - E by Extension
 - D by Date
 - S by Size
 - G Directories First
 - U Unsorted
- /A display only files with given attributes
 - display all files except with given attributes
 - D Directory
 - R Read Only
 - S System
 - H Hidden
 - A Archive

Example:

DIR /W will display the files in the current directory in wide (5-column) format.

DIR A:\DIRECT will display the files in the directory A:\DIRECT.

Echo

Purpose:

To display whether echo is on or off, set the echo state, or echo a line to the screen. Echo defaults to Off.

Usage:

ECHO [ON|OFF|<line to be echoed>]

Example:

ECHO will display whether echo is on or off.

ECHO OFF will turn echo off.

ECHO HI THERE will echo "HI THERE" to the screen.

Erase or Del

Purpose:

To erase (or delete) a file.

Usage:

ERASE <filename> [/P]
or **DEL <filename> [/P]**
/P to prompt for each file delete

Example:

DEL *.DOC /P will delete all files with the extension .DOC, and prompt the user for each file deleted.

Eset

Purpose:

To get or set Windows Command environment variables.

Usage:

ESET <variable name> <new value>

<variable name> is one of the following:

HistMinLen is the minimum number of characters a command must have to be saved into the command history. <new value> must be between 1 and 255. Default size is 3.

HistLines is the number of commands that will be saved into the command history. <new value> must be between 1 and 255. Default value is 30.

ScrBufLines is the number of lines the screen buffer will hold in memory. <new value> must be between 25 and 999. Default value is 60.

Beep determines whether the warning beep is heard. <new value> must be ON or OFF. Default is ON. This will not affect the Beep command.

Menu determines whether the menu is visible or not. <new value> must be ON or OFF. Default is ON.

Example:

ESET SCRBUFLINES 100 will set the screen buffer size to 100 lines and allow you to scroll back up 100 lines.

ESET MENU OFF will remove the menu from the top of the Windows Command window.

Except

Purpose:

To execute a command on files except for a series of specified files.

Usage:

EXCEPT (<filename>...) <command and arguments>

Example:

EXCEPT (*.DOC *.TXT) DEL *.* will delete all files in the current directory except for those with extensions .DOC and .TXT.

Exit

Purpose:

To exit Windows Command.

Usage:

EXIT

Example:

EXIT will cause Windows Command to close.

Goto

Purpose:

To branch to a label in a batch command. For more information on labels in batch files, read the [Batch Files](#) section.

Usage:

GOTO <label>

Example:

GOTO RALPH will go to the label ":Ralph" in the current batch file, and continue executing commands from that point in the file.

Help or ?

Purpose:

To access the Windows Command help file.

Usage:

HELP <topic>
or **? <topic>**

Example:

HELP TYPE will bring up the help screen for the TYPE command.

History

Purpose:

To display a history of entered commands for browsing or recalling.

Usage:

HISTORY

Example:

HISTORY will create a dialog box with all the commands held in the command history list.

If

Purpose:

Allows conditional execution of a command.

Usage:

IF [NOT] < ERRORLEVEL <number> | EXIST <filename> > <command>

NOT will make the <command> execute if the case given is not true.

ERRORLEVEL will return true if the last command exited with a return code of <number>

EXIST will return true if the file <filename> exists.

<command> is the command that will execute if the conditions are correct.

Example:

IF ERRORLEVEL 0 CLS will clear the screen only if the last command executed returned an error code of 0.

IF NOT EXIST C:\WINDOWS\WCOMMAND.EXE DIR C:\ will execute a display of the directory C:\ only if the file WCOMMAND.EXE is not found in the C:\WINDOWS directory.

Maximize or Max

Purpose:

To maximize a task. Task IDs can be found by issuing the TASK command.

Usage:

MAXIMIZE <taskid>
or **MAX <taskid>**

Example:

MAX 9337 will maximize the task with the task ID number 9337.

Mem

Purpose:

To display system statistics, free RAM, and free memory on the default drive.

Usage:

MEM

Example:

MEM will display the Windows mode, CPU type, screen resolution, free RAM and free disk space.

Minimize or Min

Purpose:

To minimize a task. Task IDs can be found by issuing the TASK command.

Usage:

MINIMIZE <taskid>
or **MIN <taskid>**

Example:

MIN 9337 will minimize the task with the task ID number 9337.

Mkdir or Md

Purpose:

To make a new directory.

Usage:

MKDIR <directory>
or **MD <directory>**

Example:

MD NEWDIR will make a subdirectory named NEWDIR in the current directory.

Move

Purpose:

To move a file.

Usage:

MOVE <fromfile> <tofile>

Example:

`MOVE *.DOC A:*.TXT` will move all files with a .DOC extension to the A drive in files with the same name, but with the extension .TXT.

Path

Purpose:

Displays the DOS Path.

Usage:

PATH

Example:

PATH will display the current Path set in DOS.

Pause

Purpose:

To prompt the user to press a key before continuing execution.

Usage:

PAUSE

Example:

PAUSE will display a prompt and wait for the user to press a key.

Print

Purpose:

To print a file through the Windows print queue.

Usage:

PRINT <filename>

Example:

PRINT *.DOC will print all files with the extension .DOC.

Prompt

Purpose:

To change the command prompt.

Usage:

PROMPT <string>

<string> is any text, which may include the following special strings:

\$p current drive and path in lower case

\$P current drive and path in upper case

\$G/\$g the ">" character

Example:

PROMPT HI \$p\$g will set the prompt to "HI " plus the current directory plus the ">" character.

Rem

Purpose:

A null, or "do nothing" command. REM can be used to echo a line to the screen if echo is on.

Usage:

REM <text>

Example:

REM HI THERE will do nothing if echo is off or write "REM HI THERE" to the screen if echo is on.

Rename or Ren

Purpose:

To rename a file.

Usage:

RENAME <filename> <newname>
or **REN <filename> <newname>**

Example:

REN *.DOC *.TXT will rename all files with a .DOC extension to a file of the same name, but with a .TXT extension.

Restore or Res

Purpose:

To restore a currently minimized task. A message is sent to the program, telling it to restore; whether or not the task actually does will depend on the application itself. Task IDs can be found by issuing the TASK command.

Usage:

RESTORE <taskid>
or **RES <taskid>**

Example:

RES 9337 will restore the application with the task ID 9337.

Rmdir or **Rd**

Purpose:

To remove an empty directory.

Usage:

RMDIR <directory>
or **RD <directory>**

Example:

RD OLDDIR will remove the directory OLDDIR from the current directory.

Runmax or +

Purpose:

To load and maximize a program.

Usage:

RUNMAX <program>
or **+ <program>**

Example:

RUNMAX THISAPP.EXE will execute the program THISAPP.EXE, and initially maximize it.

Runmin or &

Purpose:

To load and minimize a program.

Usage:

RUNMIN <program>
or **& <program>**

Example:

RUNMIN THISAPP.EXE will execute the program THISAPP.EXE, and initially minimize it.

Set

Purpose:

To display the DOS environment variables.

Usage:

SET

Example:

SET will display all the currently set DOS environment variables.

Task

Purpose:

To display all currently active tasks and their associated task IDs.

Usage:

TASK

Example:

TASK will display all the currently active tasks in a list.

Time

Purpose:

To display or set the system time.

Usage:

TIME [<hour> <minute> <second>]

Example:

TIME will display the current time.

TIME 0 50 30 will set the current time to 12:50AM and 30 seconds.

Type

Purpose:

To display a file on the screen.

Usage:

TYPE <filename>

Example:

TYPE *.DOC will display all the files with the extension .DOC on the screen.

Unalias

Purpose:

To remove a previously assigned command alias.

Usage:

UNALIAS <aliasname>

Example:

UNALIAS CDD will remove the alias CDD from the alias list, if it was previously defined.

Ver

Purpose:

To show the versions of DOS, Windows, and Windows Command currently running.

Usage:

VER

Example:

VER will display the versions of DOS, Windows and Windows Command on the screen.

Verify

Purpose:

To get or set the DOS verify flag. If verify is on, all disk writes will be verified to ensure they were written correctly. The default is off. Turning verify on will slow disk writes slightly.

Usage:

VERIFY [ON|OFF]

Example:

VERIFY will display the state of the verify flag.

VERIFY OFF will turn the verify flag off.

Vol

Purpose:

To display the current volume label on the default or specified drive.

Usage:

VOL <disk>

Example:

VOL A: will display the volume label on the disk in drive A.

Winexit

Purpose:

To exit Windows.

Usage:

WINEXIT

Example:

WINEXIT will prompt the user to verify whether or not to exit windows. If the user presses "Y", all currently running applications will be closed, and Windows will terminate.

Error Message Codes

Error Code	Error Message
10	No closing quote
20	Insufficient Memory
21	Invalid Command Name
22	Unknown Command
23	Launch Failed
30	Invalid Command Usage
31	Invalid Parameter
40	Invalid Drive
41	Insufficient disk space for
50	Path not Found
51	Invalid Path
60	Cannot Change to Directory
70	File not Found
71	File Exists
72	File Cannot be Copied to itself
79	Cannot Create Directory
80	Cannot remove current directory
81	Error removing
82	Error renaming
83	Error moving file
90	Access Denied
100	Unable to Exit Windows
110	Invalid File Size
120	Invalid Window Task ID
200	Label not Found

