WinOne©

Super Command Shell
Version 4.9
by Lucien Cinc

Introduction 2
Shareware Information 2

Summary of Main Features 3

Virus Protection 4
Command Execution Precedence 5
Command Line Macro's 5
Command Line Edit Keys 5
Tab Key Examples 6
Redirecting Command Input and Output 8
Multiple Commands 9
File Extension Association 9
ANSI Graphics control sequences 10
Wildcard Support 11

Batch Programs 11
Progman Group Icons 13
Random Desktop Wallpaper 14
Drag and Drop for files 14
Clipboard Copy and Paste 15
Screen Saver and Terminal Lock 16
Variable Font Sizes 18
User Definable Buttons 20
Status Bar 22
Unix Look and Feel 24
System Configuration 25

Command Syntax 26

Standard Commands 27
Extra Commands 34
Window Commands 44
External Commands 54
Standard Batch Commands 65
Enhanced Batch Commands 68

Introduction

Welcome, and thanks for trying WinOne !

WinOne is a Command Language Interpreter, similar in concept to the DOS shell COMMAND.COM, except that WinOne has been designed to enable you get the most out of your Windows 3.1 operating system.

WinOne provides a variety of capabilities that COMMAND.COM does not, since COMMAND.COM is not Windows aware. WinOne allows not only DOS commands and programs to be executed, but also Windows programs to be executed correctly.

WinOne attempts to make Windows easier to use, and to make you more productive when there is a need to work at the command line level. WinOne provides a rich set of commands, including addition convenience functions accessed via the System Menu. Essentially, WinOne provides a very powerful working environment, without sacrificing the flexibility and control you get from working at the command line level.

WinOne is distributed as a Commercial Shareware Program. Please read the Shareware Information section, which describes the terms and conditions of use for WinOne.

Shareware and Registration

WinOne for Windows is NOT free software. WinOne is a commercial Shareware program, which is free of any Crippleware, but contains some Annoyware. WinOne is protected by the Australian Copyright and International Copyright Laws.

WinOne can be evaluated for a trial period of 20 days. After that period, if you wish to continue using WinOne, you must register the program, otherwise, you must discontinue to use WinOne.

To register WinOne, print out the Registration Form, supplied with WinOne, in the file REGISTER.FRM, fill it out and send it along with the full registration fee to the address below :-

Lucien Cinc
56A Harbord Road
Harbord, NSW 2096
Australia

I will send you a **registration number** that will disable the displaying of DEMO in the caption bar and NOT REGISTERED in the start-up information screen, along with the **latest version** of WinOne. A registered user is entitled to **product support** via the phone (ie. both FAX and voice) and the Internet (ie. E-mail) and will also receive the next upgrade of WinOne, free of charge. Please **show your support** for the Shareware concept by registering your copy of WinOne.

Users of WinOne are encouraged to pass along the UNREGISTERED Shareware version of WinOne to other users on a trial, private non-commercial basis. WinOne may not be :-

1. Modified.
2. Distributed in a modified form.
3. Distributed in a registered state.
4. Distributed in connection with any other software, without written permission from the author.

The standard DISCLAIMER follows :-

THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

If you do not agree with the above terms and conditions you do not have permission to use WinOne, you must stop using it and remove it from your computer.

I would be happy to hear your comments on my program. Any suggestions, criticisms or bug reports, can be forwarded to the above address, or send E-mail to :-

Address: lcinc@moss.newcastle.edu.au
Subject: WinOne

Summary of Main Features

- Colour and ANSI graphics.

- Full IBM graphics character set. Use the ACS command to display the full character set.

- Full edit key functionality, including the tab key (UNIX like), which expands incomplete path, file, command or macro names currently being typed at the WinOne prompt. See Command Line Edit Keys.

- More than one command can be entered on a single line. See Multiple Commands.

- I/O redirection for WinOne commands and DOS programs. See Redirecting Command Input and Output.

- Archive file support for ZIP, LZH, ARJ, ARC. Use the ARCH command to view files inside archive files.

- File Encryption. Use the ENCODE and DECODE commands to encrypt or decrypt any file.

- Smart delete (eg. RMDIR /S and DEL / S). These commands will process sub-directories.

- Smart insertion of program files into Program Manager groups, by using the GROUP command.

- Many Extra Commands (eg. WHERE etc.).

- External Command support. External commands are simply programs that use WinOne for there input and output, giving the impression that they are executing inside WinOne. External commands are written using either Borland C or C++. See the WOIO.HLP file for more information on External Commands.

- Command line history buffer. Stores the last 30 command lines entered at the WinOne prompt. Use the HISTORY command to display a list. Also see Command Line Edit Keys.

- File Extension Associations. Associates a program(s) to a file extension. Use the EXTENSION command to manipulate file extension associations.

- Command Line Macro's. Enables sequences of commands to be grouped together. Use the MACRO command to manipulate macro's.

- Batch program support. Batch programs (ie. files having a file extension of .BAT) will be processed in the main WinOne window, unless the CALL command is used to run the batch program, then the batch program will be executed under DOS.

- Extended Wildcard Support. Wildcard characters question mark and star (ie. ' ? ' and ' * ') can be placed anywhere inside a file name.

- File and directory descriptions, up to 60 characters. See the DESCRIBE command.

- One pass floppy diskette support, using the TOFILE and TODISK commands.

- User Definable Buttons. There are a maximum of seventeen user definable buttons, that are fully programmable. Use the **Buttons** option in the system menu to manipulate buttons..

- 43 selectable button images.

- A screen buffer, which can store up to sixteen screens in memory via a vertical scroll bar.

- Clipboard copy and paste.

- Variable Font Sizes. Set different font sizes for the WinOne Window. Use the **Fonts...** option in the system menu to change font sizes.

- Built-in Screen Saver and Terminal Lock. Use the **Lock setup...** option in the system menu.

- Easy access to the configuration files, AUTOEXEC.BAT, CONFIG.SYS and all .INI files. Use the **System Edit...** option in the system menu. Also see System Configuration files.

- Auto Window Raise. Automatically raises a window that contains the mouse cursor, without the need to press the left mouse button. See the RAISE command.

- Status Bar. Displays time, bytes free and a percentage done indicator for currently executing commands.

- Custom Program Manager Group Icons that replace default group icons.

- Random Desktop Wallpaper, which changes the wallpaper every time WinOne is run.

- Drag and Drop for files either dropped to WinOne or dragged (ie. using command DRAG) from WinOne.

- Comprehensive user manuals supplied in Word for Windows 2.0a format (ie. WIN_ONE.DOC and WOIO.DOC) and in postscript format (ie. WIN_ONE.PS and WOIO.PS).

Virus Protection

When WinOne is modified in any way, a window is displayed, informing the user that WinOne has been modified and it could be a virus. WinOne will then terminate :-

Command Execution and Precedence

Internal commands have the highest priority, otherwise when a command is entered with no extension, then the extensions .COM, .EXE, and .BAT are tried, in that order. The PATH environment variable is used to locate an external command or program.

When an extension is specified and the file can not be located, then the error message **Bad command or file name** is displayed. WinOne will NOT prompt the user to insert the file into A:.

All DOS programs (including commands) can be executed from the WinOne prompt. When a DOS program is executed then the WinOne prompt will not appear until the program has completely finished executing.

The user can place an executing DOS program in the background before it has completed executing, and thereby displaying a new WinOne prompt , by pressing the control key and the z key together (ie. CTRL Z). This feature works best when running Windows in Protected Mode or 386 enhanced mode, along with the appropriate PIF settings, so that all DOS programs appear inside a window, instead of full screen.

Windows programs behave differently to DOS programs. When executing a Windows program the WinOne prompt will re-appear straight away, ready for the next command and will not wait until the program has finished executing. Also many Windows programs allow command line arguments to be past on the command line, similarly to normal DOS programs and commands. For example, to edit a file called TEST.TXT, enter at the WinOne prompt :-

NOTEPAD TEST.TXT

Command Line Macro's

A Command line Macro enables a new command to be created, from an original command or from a series of original commands. Command line Macro's can be created, deleted and listed using the MACRO command.

Command Line Edit Keys

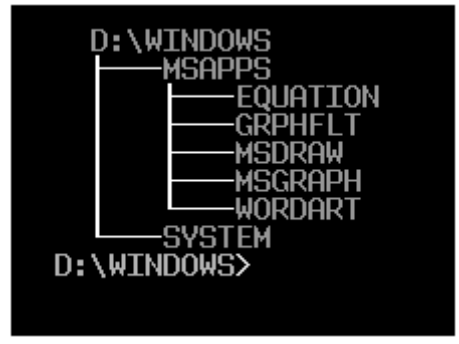
Key	Function
left arrow	Move one character left.
right arrow	Move one character right.
up arrow	Display the previous command line.
down arrow	Display the next command line.
ctrl up arrow	Display the oldest command line.
ctrl down arrow	Display the last command line.
ctrl left arrow	Move one word left.
ctrl right arrow	Move one word right.
del	Delete one character forward.
backspace	Delete one character backward.
home	Move to beginning of line.
end	Move to end of line.
page up	Scroll up a page.
page down	Scroll down a page.
ctrl page up	Scroll up a line.
ctrl page down	Scroll down a line.
ctrl insert	Copy to clipboard.
shift insert	Paste from clipboard.
escape	Clear the line.
tab	Expand an incomplete path, file, command or macro name that is currently being typed at the WinOne prompt.
ctrl tab	Display a list of names that can complete the current incomplete path, file, command or macro, that is being typed at the WinOne prompt.

Note:
Insert mode is always on.

Also see, examples on using the Tab key.

Tab Key Example

The Tab key will expand an incomplete path, file, command or macro name, that is currently being typed at the WinOne prompt. For example, consider the following tree structure and default directory :-



To change to the MSAPPS sub-directory, enter at the WinOne prompt :-

CD MS<tab>

where <tab> is the tab key.

The new command line will be :-

CD MSAPPS

When there is more than one name, that can be substituted, then WinOne will beep, to let the user know. Press the Tab key again to select the next name. For example, assume the default directory is D:\WINDOWS\MSAPPS. To change to the sub-directory MSGRAPH, enter at the WinOne prompt :-

CD MS<tab><tab>

After the first Tab is pressed, the command line will be :-

CD MSDRAW

After the second Tab is pressed, the command line will be :-

CD MSGRAPH

Similarly, file, command and macro names can be expanding in the same way.

Press the Control key along with the Tab key (ie. CRTL TAB) to display a listing of all the names found. The list is sorted alphabetically with command and macro names first, then directory names and finally file names. For example, consider :-

Adding command search paths

Since commands can include any program names (ie. file names that have an extension of .EXE, .COM or .BAT), a search path can be added to the WIN_ONE.INI file, so that, every time

the Tab key is pressed, the directories specified will be searched and any suitable program names that are located will be added to the command names. When a program name is located the file extension is removed from the file name before it is added to the list of commands. The search path is added in the following way :-

1. Use Notepad to edit the WIN_ONE.INI file.
2. In the WinOne section, Insert the line :-

Path=path1;path2;path3...

path1, path2, etc any valid full path.

For example, to add all the commands in the directories C:\DOS and D:\WINDOWS, insert the line **Path=c:\dos;d:\windows**, as below :-

```
[WinOne]
Path=c:\dos;d:\windows
FontHeight=14
FontWidth=9
XOrigin=47
YOrigin=43
ScreenWidth=80
ScreenHeight=24
ColourMode=on
DisplayMacro=off
```

3. Save and exit notepad.

Since, the search path is independent of the PATH environment variable, which is used to locate a program to execute, any paths that are specified, should also be included in the PATH environment variable, otherwise WinOne will not be able to locate the program to execute after it has been expanded.

Now commands such as FORMAT can be expanded using the tab key, for example, enter at the WinOne prompt :-

FORM<tab>

Note:
Directory names will have backslash character added to the end of the name, and all other names will have a space character added to the end of the name.

Redirecting Command Input and Output

WinOne allows Standard Input and Standard Output to be redirected on the command line and assumes that Standard Input comes from the keyboard and that Standard Output goes to the screen. The default Standard Input and Standard Output can be redirected using the following characters on the command line :-

- The less than sign (ie. ' < ') to use the contents of a file as the input for a command. For example :-

SORT < filename

- The greater than sign (ie. ' > ') to send the output from a command to a file. When the specified file does not exist, it is created and when the file does exist, it is over written, and the previous contents are lost. For example :-

```
SORT  >  filename
```

- The double greater than sign (ie. ' >> ') appends the output from a command to the end of a file.

- The bar character (ie. ' | ') allows the Standard Output from one command to be used as the Standard Input to another command. This is referred to as a pipe. For example :-

```
NAME1  |  NAME2  |  NAME3  ...
```

The commands NAME1, NAME2 and NAME3 are executed in that order, that is from left to right.

Note:
When using redirection with DOS programs, executed tasks can not be placed in the background. For more information on placing DOS tasks in the background see Command Execution and Precedence.

Multiple Commands

WinOne allows more than on command to be entered on the command line. Simply separate commands by a CRTL T character (ie. press the control key and the T key together).

The default CRTL T separator character can be changed to any printable character in the following way :-

1. Use Notepad to edit the WIN_ONE.INI file.
2. In the WinOne section, Insert the line **CharSplit=value**, where value is the ASCII character code, as below :-

```
[WinOne]
FontHeight=14
FontWidth=9
XOrigin=10
YOrigin=52
ScreenWidth=80
ScreenHeight=39
ColourMode=on
CharSplit=59
DisplayMacro=off
Path=c:\dos;d:\windows
```

3. Save and exit notepad.
4. Exit WinOne.
5. Re-run WinOne, so that the change can take effect.

For a complete list of ASCII character codes see the ACS command.

File Extension Associations

Filename associations allow the user to enter file names, other then .EXE, .COM, or .BAT, at the WinOne prompt and provided that an association exists for the file extension, the respective program will be executed, with the filename past as a parameter.

For example, an association can be created for all .DOC files that will execute the word-processor WINWORD.EXE. To create or delete associations use the EXTENSION command.

Similarly, a single file extension may be associated with many programs. The user will be prompted to select one of the programs to execute. For example, assume an association exists between .DOC files and the three programs EDIT.COM, NOTEPAD.EXE and WINWORD.EXE. The following window is displayed :-

Simply, use the mouse and click on the button to execute the desired program.

Not all associations need to be created by the user. Many Windows programs automatically create associations for you, however DOS programs do not, and they need to be created manually using the EXTENSION command.

ANSI Graphics Control Sequences

ANSI Control sequences are combinations of characters that can be used to control the cursor, screen and keyboard. The following ANSI Control sequences are supported by WinOne :-

Cursor Movement

ESC[H	Home cursor.
ESC[<a>;<b>H	Move cursor to <a>, <b>.
ESC[<a>;<b>f	Same as for ESC[<a>;<b>H
ESC[<a>A	Move cursor up <a> spaces.
ESC[<a>B	Move cursor down <a> spaces.
ESC[<a>C	Move cursor right <a> spaces.
ESC[<a>D	Move cursor left <a> spaces.
ESC[s	Save cursor position
ESC[u	Restore cursor position

Screen Control

ESC[2J	Clear screen

ESC[K	Erase to end of line
ESC[<n>;...;<n>m	Activate given attributes, where each <n> is 0 to 8 Sequence is Ignored 3x Foreground colour x 4x Background colour x where x is 0 Black 1 Red 2 Green 3 Yellow 4 Blue 5 Magenta 6 Cyan 7 White
ESC[=<n>h	NOT Supported
ESC[=<n>l	NOT Supported

Keyboard Control

ESC[<ns>;...;<ns>p	NOT Supported

Note:
ANSI Control sequences can be entered via the PROMPT command, or by displaying a file that already contains the codes, using the TYPE command.

ANSI Control sequences can NOT be entered at the command line prompt.

Wildcard Support

WinOne supports both star and question mark characters (ie. ' * ' and ' ? ') as Wildcard characters,

WinOne recognises all the normal Wildcard formats, as well as some **additional formats** that can be used for all internal commands.

Normal Wildcards:

1. ***.***	All files.
2. ***.EXE**	All files that have an extension of 'EXE'.
3. **WIN*.***	File names beginning with 'WIN'.
4. **WIN*.E***	File names beginning with 'WIN' and an extension beginning with 'E'.

additional Wildcards:

5. ***ONE.***	File names ending with 'ONE'.
6. ***ONE.*E**	File names ending with 'ONE' and an extension ending with 'E'.
7. ***O*.***	File names that contains 'O'.
8. ***O*.*X***	File names that contains 'O' and an extension that contains an 'X'.
9. ***IN_O*.***	File names that contains 'IN_O'.
10. ***IN_O*.*E***	File names that contains 'IN_O' and an extension that contains 'E'.
11. **W*E.***	File names that starts with 'W' and ends with 'E'.
12. **WI*NE.***	File names that starts with 'WI' and ends with 'NE'.
13. **WI*_*NE.***	File names that starts with 'WI' and ends with 'NE' and contains '_'.

All the above formats can be used to specify a file called 'WIN_ONE.EXE'

Note:

Essentially Wildcard characters can be place anywhere inside a filename, and still be interpreted correctly by all WinOne commands.

Do NOT use the additional Wildcard formats with any programs (eg. when executing a program, either DOS or Windows), since it is the responsibility of the individual program to process the Wildcards, they may be incorrectly interpreted.

Batch Programs

A batch program is a simple text file that contains a sequence of WinOne or DOS commands. The NOTEPAD text editor can be used to create a batch program. Batch programs will be displayed inside the main WinOne window, unless the CALL command is used to run the batch program, then the batch program will be displayed in either another window or full screen, depending on the default PIF setting.

Standard Batch Commands

CALL	Run a second batch program, then return to the first batch program.
ECHO	Display a message or turns echo on or off.
FOR	Perform a command for each file in the specified set of files.
GOTO	Switch to another part of the batch program, and continue executing the program from that point.
IF	Perform conditional processing in a batch program.
PAUSE	Suspend processing of a batch program and display a message.
REM	Allows comments inside a batch file.
SHIFT	Change the position of replaceable parameters in a batch program.

Enhanced Batch Commands

ASK	Ask a yes/no question and set the errorlevel respectively.
BEEP	Send a beep to the system speaker.
BOX	Display a box in one of four pre-defined formats.
CALC	Perform basic arithmetic calculations in a batch program. The result is stored in an environment variable.
COLOUR	Change foreground and background screen colours.
DIRS	Display the directory stack.
END	End a batch program.
GETKEY	Wait for a single keypress from the user. The character is stored in an environment variable.
GETSTR	Wait for a sequence of keypresses from the user. The sequence of characters are stored in an environment variable as a string of characters.
GOSUB	Jump to another part of a batch program, and continue executing from that point until RETURN is encounted.
LOCATE	Position the cursor anywhere on the screen.
LOWER	Convert a text string to lower case. The converted text string is stored in an environment variable.
PARSE	Allows a sentence to be broken into pieces. The pieces are stored in environment variables.
POPD	Pop a directory from the directory stack and make this directory the current directory.
PUSHD	Push the current directory onto the directory stack and change to the specified directory.
RETURN	Return execution to the next command following the GOSUB command.
SAY	Display a message. This command with not add a carriage return - line feed at the end of the message.
SLEEP	Do nothing for a time.
STOP	Stop processing a batch program and continue processing the batch program that called this one.

STRSIZE	Determine the length of a string. The length is stored in an environment variable.
SUBSTR	Extract a section of text from a text string. The extracted text string is stored in an environment variable.
UPPER	Convert a text string to upper case. The converted text string is stored in an environment variable.

Replaceable Parameters

Batch programs can be past arguments on the command line. These arguments can be referenced by using replaceable parameters. There are a maximum of 10 replaceable parameters allowed, specified by %0 through to %9 inside a batch program. For example, consider the batch program called MV.BAT, which will move all the files from one directory to another directory :-

```
COPY %1\*.*  %2
DEL %1
```

To move all the files from C:\FIRST to the directory C:\SECOND, Enter at the WinOne prompt :-

```
MV  C:\FIRST C:\SECOND
```

Environment variables

Environment variables can also be used inside batch programs and have the following format %name%. For example, %COMSPEC% inside a batch program will be replaced with C:\COMMAND.COM.

Program Manager Group Icons

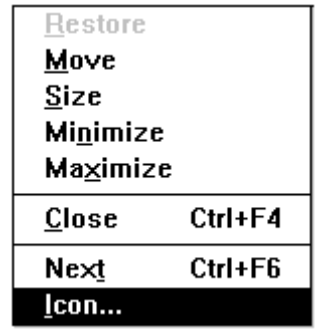
WinOne allows you to visually customise your Windows Program Manager group icons by replacing the default Program Manager group icons with new icons :-

The icons inside any file with an extension of .EXE, .DLL or .ICO can be used to replace the default Program Manager Group icons.

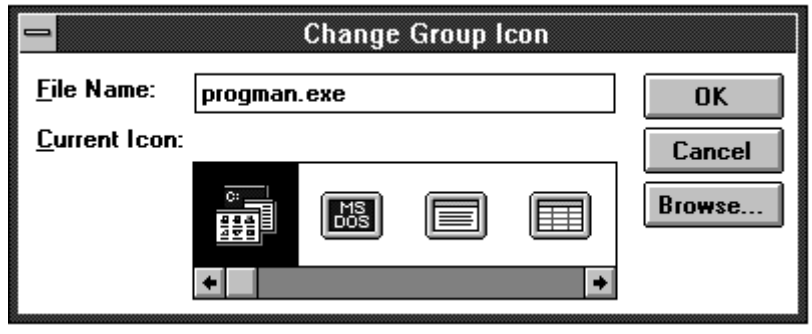
This feature is not automatically enabled, it must be enable or disabled using the GROUP command :-

GROUP ON	- to enable and
GROUP OFF	- to disable.

Once enabled, simply click once on the System Menu Box (ie. in the upper left corner of the window) of any Program Manager Group. This will open that group's System Menu. You will notice that a new menu option **Icon...** has been added to the Groups System Menu :-



When this option is selected the following window appears :-



The default PROGMAN.EXE filename can be changed to any of the above file types (ie. .EXE, .DLL or .ICO). Either enter a path for the file containing the icons or press the Browse button, which will display the Open window, then select the path for the icons. Having selected a path, then choose the desired icon from the icons displayed inside the list box, and press the OK button to set the Group icon to the newly selected icon.

When WinOne is running and this feature is enabled, the Program Manager Groups will display the new icons.

Random Desktop Wallpaper

The Desktop Wallpaper can be changed every time WinOne is executed. WinOne will select a new bitmap (ie. .BMP file) from a list, to set as the new wallpaper next time Windows is run.

The list of bitmap files is specified in the following way :-

1. Use Notepad to edit the WIN_ONE.INI file.
2. Insert a new section

```
[Wallpaper]
wall0=drive:\path\filename.BMP
wall1=drive:\path\filename.BMP
        '           '
        '           '
        '           '
wall19=drive:\path\filename.BMP
```

3. Save and exit notepad.

This feature is only enabled when the above section contains one or more valid bitmap files and invalid bitmap files will result in no changes to the current wallpaper.

Also see command WALLPAPER to manually set the Desktop Wallpaper.

File Drag and Drop

WinOne supports Drag and Drop for any program that can drag files (eg. the File Manager). One or more file or path names can be dragged into the main WinOne window, and when dropped, the file or path names will be added to the end of the WinOne command line.

For example, to drag a file from the File Manager, simply position the mouse cursor over the file to drag, press and hold down the left mouse button, drag the file until the mouse cursor is inside the main WinOne window and release the left mouse button to drop the file. During the dragging process the cursor will change to one of the following, as it is moved around the desktop :-

the window can accept the file.

the window can accept the files.

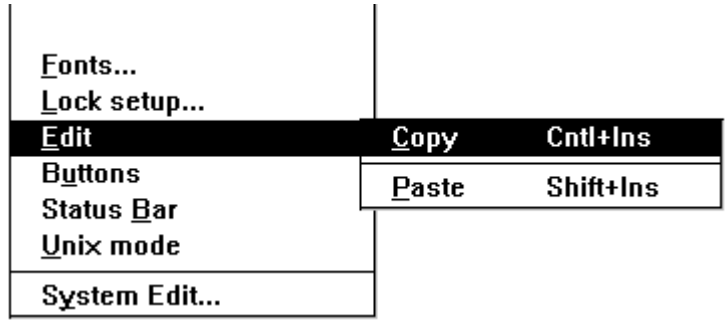
the window can not accept the file or files.

File names that are dropped into the main WinOne window, will have a space character added to the end of the file name, and path names will have a backslash character added to the end of the path name.

To drag one or more files from WinOne to another program that accepts a file drop (eg. the Program Manager) use the DRAG command.

Clipboard Copy and Paste

The system menu contains the **Edit** option, which when selected will display a sub-menu containing the options **Copy** and **Paste**, as below :-



Marking a region

Use the mouse cursor and position it over the first character to mark. Press and hold down the left mouse button. While holding down the left mouse button, drag the cursor over the last

character to mark and release the left mouse button. The marked region will now be displayed in reverse colours.

Similarly, to mark a word, simply position the mouse cursor over the word and double click the left mouse button.

To clear a region that is marked on the screen, press the left mouse button once.

WinOne allows a region to be marked and copied to the clipboard, whenever WinOne is waiting for a key stroke or a series of key strokes to be entered.

Copying to the clipboard

Having marked a region on the screen, there are several ways to copy it to the clipboard :-

1. Select the **Edit** option in the system menu, and then select **Copy**.
2. Press the Control key and the Insert key together (ie. CRTL INS).
3. Press the right mouse button.

When a region is successfully copied to the clipboard the highlight is cleared from the screen.

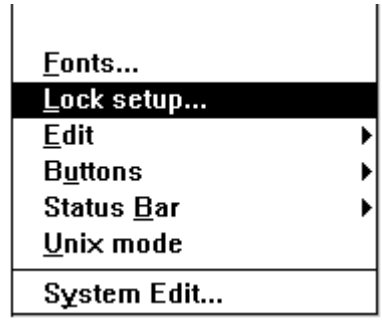
Pasting from the clipboard

Text that has been copied to the Clipboard can be pasted to the command line or any command that is waiting for string input. There are several ways to paste from the clipboard :-

1. Select the **Edit** option in the system menu, and then select **Paste**.
2. Press the Shift key and the Insert key together (ie. SHIFT INS).
3. Press the middle mouse button.
4. Double click the right mouse button. When a region is marked on the screen, the region is first copied to the clipboard, then the clipboard contents are pasted to the command line, otherwise, when there is no region marked on the screen, only the clipboard contents are pasted to the command line.

Screen Saver and Terminal Lock

The system menu contains the **Lock Setup...** option, as below :-



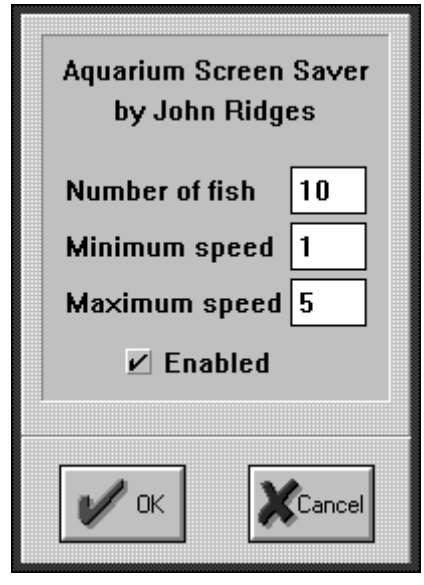
When selected the Terminal Lock Setup window is displayed :-

SPX module Configuration

The Config list box contains the names of all the SPX modules that can be configured.

To configure any of these modules, use the mouse and DOUBLE click on the name inside the list box. This will display the individual configuration window for that SPX module.

Each individual configuration window will contain the author of the module, an Enable check box, and any variable parameters that can be set (eg. speed, etc) :-



To enable a module, use the mouse and click in the Enable check box and a small tick mark will be displayed in the box. To disable the module, simply click in the Enable check box again and the tick mark will be removed.

When a module is enabled, a small circle is also added to the end of the modules name, inside the main Config list box.

Delay in minutes

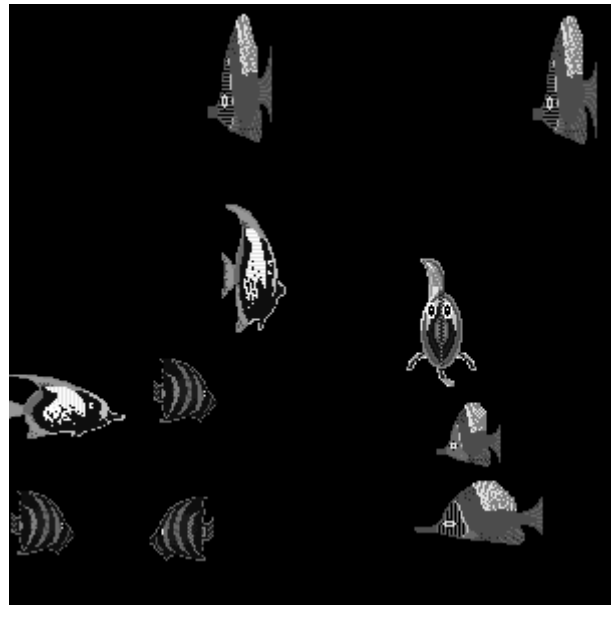
Specifies the length of time, with no keyboard input or mouse movement, that must elapse before the screen saver is activated. When the screen saver is activated, WinOne will randomly select one of the enabled SPX modules and display it.

Enable Screen Saver
Switches the screen saver on or off.

Password
The Use Password check box, determines whether or not to prompt the user to enter a password before returning control to Windows.

When the password is enabled, another window will be displayed, requesting a new password :-

View button
Press the View button to see a sample of the SPX module, that is highlighted in the main Config list box :-



Hot Spots
A Hot Spot is an area on the desktop where the mouse cursor can be placed to signal to the screen saver to either :-

1. Never activate the screen saver, by leaving the mouse cursor in the top left hand corner of the desktop.

2. Active the screen saver, by leaving the mouse cursor in the top right hand corner of the desktop for 2 seconds.

Acknowledgments

The SPX modules have been written by a number of different and very talented authors. The modules were originally written for Screen Peace, which is an excellent screen saver. Screen Peace and all the SPX modules are all free public domain. The information necessary to write your own SPX module is available from the Coast Communication BBS.

Note:

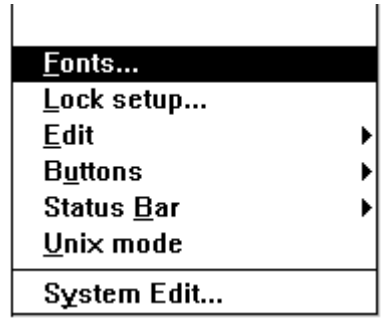
Also see the LOCK command to activate the screen saver from the command line..

All the SPX modules must be located in the same directory as the WinOne executable file WIN_ONE.EXE.

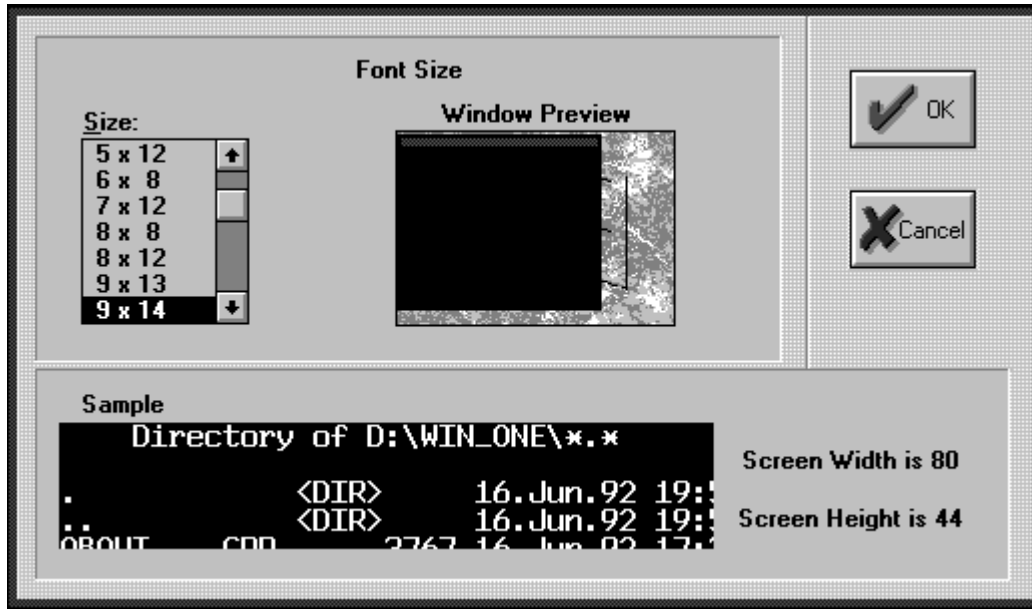
Since WinOne allows other tasks to multi-task, DISABLE all other screen savers before using this screen saver or terminal lock. However, the Windows Screen Saver does not need to be disabled, WinOne automatically disables it on start up and re-enables it on exit.

Variable Font Sizes

The system menu contains the **Fonts...** option, as below :-



When selected a window will be displayed, containing a list of font sizes, a window preview, showing the WinOne window size with respect to the desktop, and a sample of the selected font size :-



Simply use the mouse to select a font size in the Size list box.

There are 10 standard font sizes, that are supplied with Windows :-

4 x 6
5 x 12
6 x 8
7 x 12
8 x 8
8 x 12
10 x 18
12 x 16
16 x 8
16 x 12

Additional Fonts:

WinOne allows a maximum of 7 additional fonts to be added to the above lists of fonts. Only fonts that are for Windows DOS applications (ie. Terminal font types) can be added.

Fonts are installed in the following way :-

1. Use Notepad to edit the WIN_ONE.INI file in the Windows directory
2. Insert a new section :-

```
[Fonts]
font0=filename.FON
font1=filename.FON
      '          '
      '          '
      '          '
font7=filename.FON
```

where *filename* refers to the font file name.

3. Save and exit Notepad.
4. Copy the font files into the Windows SYSTEM sub-directory.
5. Exit WinOne and re-run, so that the changes can take effect.

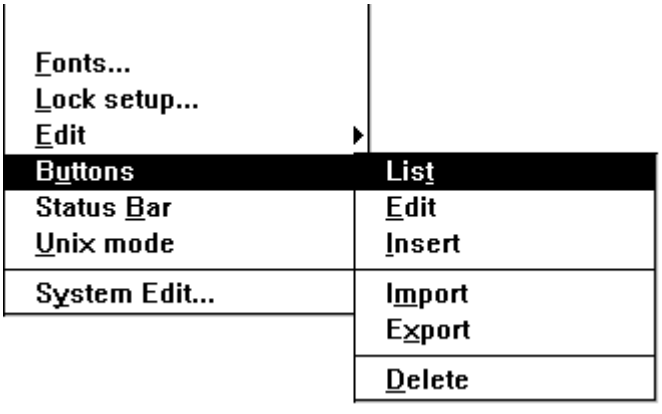
6. Use the **Fonts...** option from the system menu to display and select the new fonts.

The above picture shows two addition font sizes (ie. 9 x 13 and 9 x 14). These and many more different font sizes are available from the Coastal Communication BBS.

User Definable Buttons

User Definable Buttons enable the user to associate a command line with a button. When the button is pressed then the specified button command line is executed. There are three system buttons that CANNOT be modified in any way (ie. the red Exit button, the blue Shell button and the green Help button), and seventeen additional user buttons that can be modified (ie. the yellow buttons) :-

The system menu contains the **Buttons** option, which when selected will bring up a sub-menu, as below :-



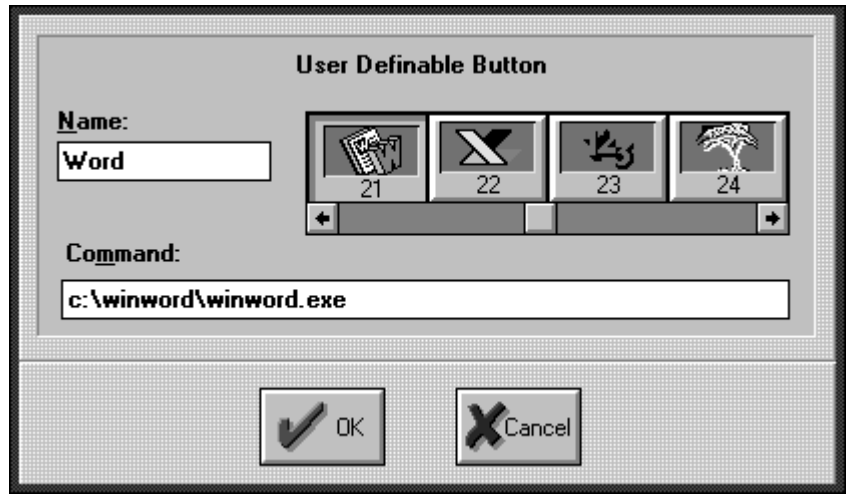
The sub-menu contains the following options :-

1. List - Displays a list of the commands associated with each button.
2. Edit - Modify an existing button.
3. Insert - Add a new button.
4. Import - Add or import a new button image.
5. Export - Delete an imported button image.
6. Delete - Remove an existing button.

When editing, inserting or deleting a button, a message is displayed , informing the user that the buttons have been marked to perform the specified function. Also, the cursor will be changed to a hand cursor. Then the next time a button is pressed, using the mouse, the specified function will be carried out. Type ^C at the WinOne prompt to cancel the operation.

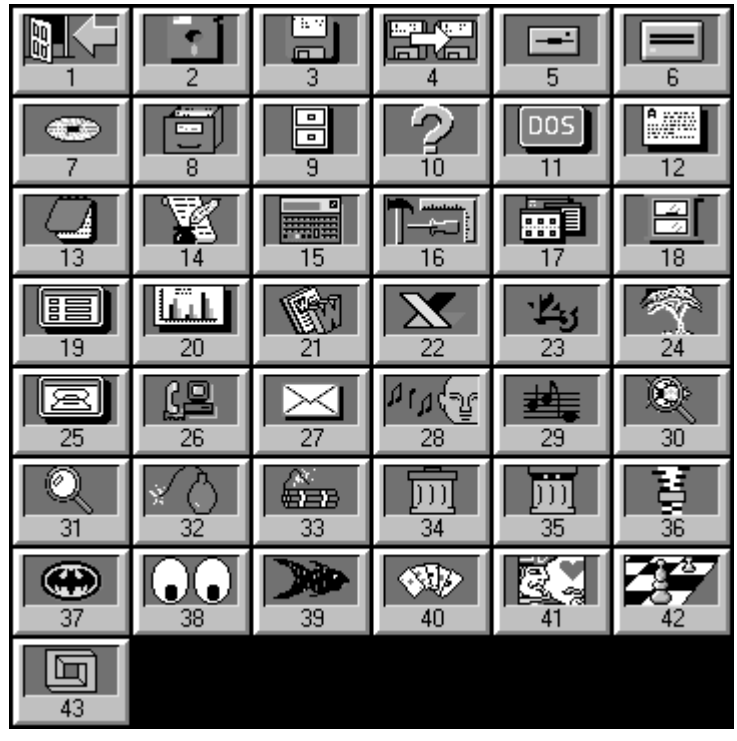
When editing or inserting a button, a window will appear, where the button name, command line and the image for the new button is specified. The images displayed includes all the default

images, as well as the imported images The button command line can include anything that can be entered at the WinOne prompt, this includes File Extension Associations and Macro's.

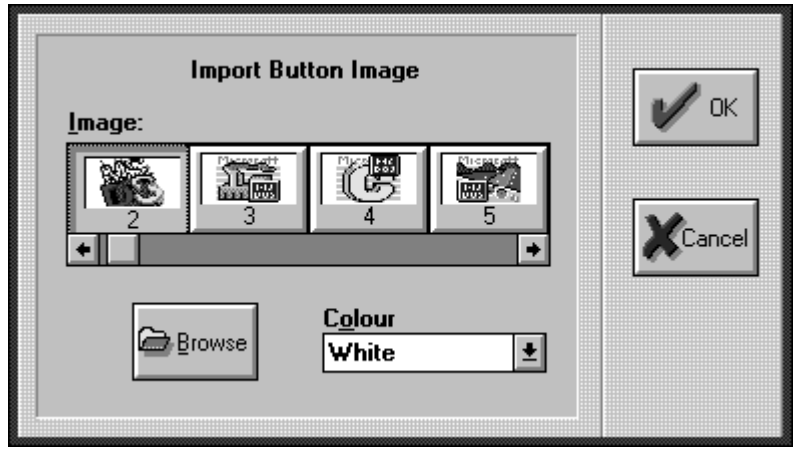


Button Images

There are 43 default images that can be displayed inside a button, as follows :-



Importing button images allows custom images to be displayed inside a button. Images are created from icons that are stored inside files that have an extension of .EXE, .DLL or .ICO. The following window is displayed when importing a button image :-



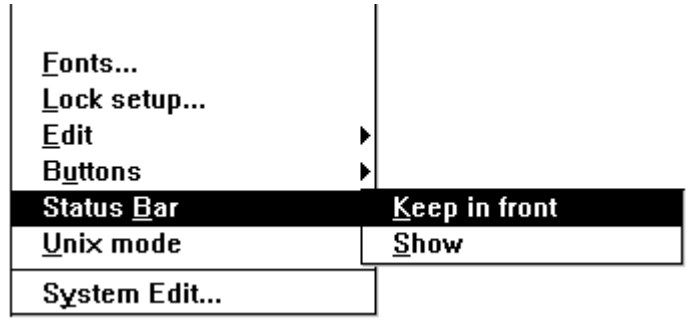
Press the Browse button to select a file to extract the images from. After a file has been selected then all the images inside the file will be displayed inside the list box. Simply select the image to import from this list box and press the OK button. The image will then be added to the end of all the default button images.

To remove an imported button image select export from the button menu. The following window is displayed ;-

Simply select the image to export from the list box and press the OK button.

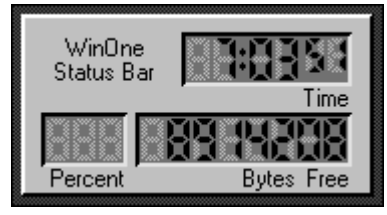
Status Bar

The system menu contains the **Status Bar** option. When selected a sub-menu is displayed with the options **Keep in front** and either **Show** or **Hide**. :-



Displaying the Status Bar

The option **Show** will display the Status Bar, showing the current time, the amount of free bytes in the global memory heap, and a percent indicator, which displays how much of a command is done. When no command is executing the percent display is blank.

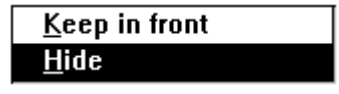


To move the Status Bar, simply position the mouse cursor inside the Status Bar window, hold down the left mouse button, then drag the white rectangle to its new position and release the left mouse button.

Not all the WinOne Commands use the Percent indicator, For example, CLS, does not.

Hiding the Status Bar

The option **Hide** removes the Status Bar :-

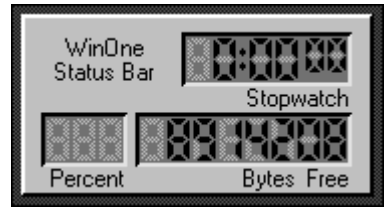


Keeping the Status Bar in front

The option **Keep in front** will keep the Status Bar in front of all other Windows, even when the Status Bar is not active. This option is used to toggle this feature on and off. A tick mark will be placed next to the option text when it is on :-

Stopwatch

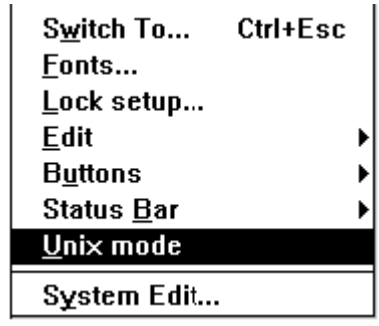
To either display or remove the Stopwatch simply click the right mouse button, while the mouse cursor is inside the Status Bar. Clicking the right mouse button twice, while the time is displayed, will automatically start the Stopwatch.



Pressing the SPACE key will either start or stop the Stopwatch and pressing the ESC key will reset the Stopwatch back to 0:00 00.

Unix Look and Feel

The system menu contains the **Unix mode** option, as below :-



Selecting the **Unix mode** option will either switch this mode off, signified by no tick mark next to the option text, or switch this mode on, signified by a tick mark next to the option text :-

Switching this option on will give WinOne a look and feel similar to the Unix Operating System.

Unix Paths

Path names under Unix use front slash characters (ie'/') instead of back slash characters (ie. '\') to separate the directory names. For example, the Windows system directory is specified in the following way :-

D:/WINDOWS/SYSTEM

Unix Command Line Switches

Command line switches under Unix are signalled using a minus sign character (ie. '-') instead of a front slash character (ie. '/'). Also, a space character must proceed the minus sign character for the command line switch to be correctly interpreted. For example, to display a wide sorted directory listing of all the files in the Windows system directory, enter at the WinOne prompt :-

DIR D:/WINDOWS/SYSTEM/*.* -OW

There are some special cases where a space character followed by a minus sign character should NOT be interpreted as a command line switch, but instead be interpreted simply as a space character and a minus sign character. For example, to remove the Read Only attribute for all the files in the current directory :-

ATTRIB -R *.*

under Unix, will be incorrectly interpreted as :-

ATTRIB /R *.*

In this case, simply add an extra minus sign character. WinOne will interpreter a space character followed by two minus sign characters as simply a space character and a minus sign character :-

ATTRIB --R *.*

Similarly, DOS programs that already use minus sign characters to specify command line switches, will also need an extra minus sign character inserted to be correctly interpreted under Unix. For example, to test the contents of a ZIP file, under Unix, enter at the WinOne prompt :-

UNZIP --T SOMEFILE.ZIP

Command line strings

There is no difference between strings specified under Unix or DOS.

Unix Output

All output is displayed according to the above rules under Unix. Additionally, Unix path names will be displayed in lower case characters, where ever possible. The lower case mapping of path names can, however, be over-ridded, in the following way :-

1. Use Notepad to edit the WIN_ONE.INI file in the Windows directory
2. In the UnixMode section, Insert the line **Case=type**, where type is either upper (ie. for upper case characters), or lower (ie. for lower case characters), as below :-

```
[UnixMode]
Unix=1
Case=upper
```

3. Save the changes to WIN_ONE.INI
4. Exit WinOne and re-run, so that the changes can take effect.

Displaying DOS equivalent commands

DOS equivalent of Unix commands can be set to display after each command is entered, in the following way :-

1. Use Notepad to edit the WIN_ONE.INI file in the Windows directory
2. In the UnixMode section, Insert the line **Display=value**, where value is either 1 (ie. display DOS equivalent commands), or 0 (ie. do not display DOS equivalent commands), as below :-

```
[UnixMode]
Unix=1
Display=1
Case=upper
```

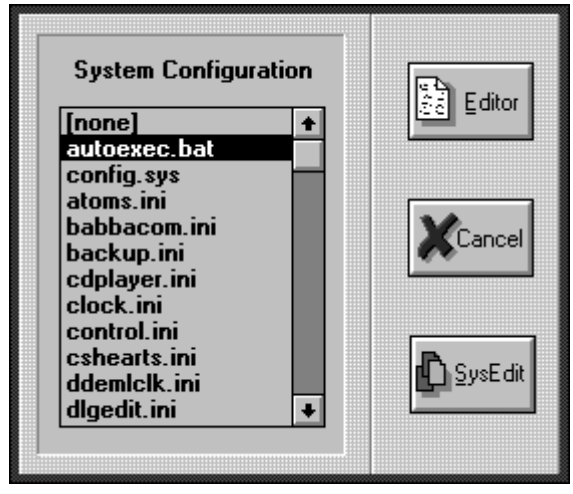
3. Save the changes to WIN_ONE.INI
4. Exit WinOne and re-run, so that the changes can take effect.

System Configuration files

The system menu contains the **System Edit...** option, as below :-

When selected a window is displayed, showing a list of all the configuration files. These files include :-

1. AUTOEXEC.BAT
2. CONFIG.SYS
3. all .INI files



Use the mouse, to select the file to edit and press the Editor button.

SysEdit Button
Windows already includes a System Configuration editor for the following files :-

1. AUTOEXEC.BAT
2. CONFIG.SYS
3. WIN.INI
4. SYSTEM.INI

Use the SysEdit button to quickly edit any of the above files.

Note:
Do NOT alter any of the configuration files, when in doubt.

Command Syntax

Consider the following command syntax :-

NAME [drive:][path][filename] [...] [ON | OFF] variables /ABC

Words in blue upper case letters are keywords, and must be entered to specify the command.

Words and characters in magenta make up the rest of the command line. Usually referred to as the command line tail. Lower case words represent variables.

The variable drive: specifies the disk drive. For example A:, B:, C:, etc.

The variable path specifies a DOS path name.

The variable filename specifies a DOS file name.

Words or letters in square brackets (ie. ' [' and ']') represent optional parameters and they do not need to be specified to execute the command.

Three periods (ie. ' ... ') suggests that additional parameters are allowed be not necessary.

Words of letters separated by a vertical bar (ie. ' | ') represents one selection from a list.

A group of letters preceded by a front slash (ie. ' / ') specifies one or more switches . Switches represent command options that can be specified.

Command Argument Strings

A String is a sequence of characters enclosed in double-quote marks. For example, consider the string *"Hello, world"*.

To specify a double-quote mark inside the string, enter two double-quote marks. For example, the string *"Some ""quote marks"" inside a string"* will be interpreted as *Some "quote marks" inside a string*.

Standard Commands

Command ATTRIB

Function:

Displays or changes file attributes.

Syntax:

ATTRIB [+R | -R] [+A | -A] [+S | -S] [+H | -H] [[drive:][path]filename] [/S]

+	Sets an attribute.
-	Clears an attribute.
R	Read-only file attribute.
A	Archive file attribute.
S	System file attribute.
H	Hidden file attribute.
/S	Process sub-directories.

Command CHDIR or CD

Function:

Displays the name of or changes the current directory.

Syntax:

CHDIR [drive:][path]
CD [drive:][path]

Note:

To easily move between nested sub-directories see command GO.

Also see command CDD

Type CD drive: to display the current directory in the specified drive
.

Type CD without parameters to display the current directory.

Command CLS

Function:

Clears the screen.

Syntax:

CLS [/B]

/B	Clear the internal screen buffer.

Command COPY

Function:

Copies one or more files to another location.

Syntax:

COPY [/A | /B] source [destination] [/V][/Q]

source	Specifies the file(s) to be copied.
destination	Specifies the directory and/or filename for the new file(s).
/A	Ignored.
/B	Ignored.
/Q	Quiet Mode. Only Error Messages are displayed.
/V	Verifies that new files are written correctly.

Note:

All files are opened in binary mode.

The current directory is used when the parameter destination is not specified.

COPY does not support appending of files.

Also see the commands MOVE, SMOVE and SCOPY.

Command DATE

Function:

Displays or sets the date.

Syntax:

DATE [date]

date dd-mm-yyyy

Note:

Type DATE without parameters to display the current date setting.

Command DEL or ERASE

Function:

Deletes one or more files.

Syntax:

DEL [drive:][path]filename [/P] [/S] [/Y]
ERASE [drive:][path]filename [/P] [/S] [/Y]

[drive:][path]filename Specifies the file(s) to delete.
/P Prompts for confirmation before deleting each file.
/S Process sub-directories.
/Y Assume YES for all questions. Questions are not displayed.

Note:

Also see command DELBUT.

Command DIR

Function:

Displays a sorted list of files and sub-directories in a directory.

Syntax:

DIR [drive:][path][filename] [/P] [/W] [/A[[:]attributes]] [/O] [/S] [/L]

[drive:][path][filename] Specifies drive, directory, and/or files to list.
/P Ignored.
/W Uses wide list format.
/A Displays files with specified attributes. where attributes are
- D Directories
- R Read-only files
- H Hidden files
- A Archive files
- S System files

-	Prefix meaning NOT
/O	Display file names in alphabetical order.
/S	Process sub-directories.
/L	Display file names in lower case.

Note:

Switches that are preset in the DIRCMD environment variable are ignored.

Command EXIT

Function:

Exit WinOne or Windows.

Syntax:

EXIT [/W] [/Q] [/R] [/B]

/W	Exit Windows via the Program Manager.
/Q	Exit Windows quickly.
/R	Restart Windows.
/B	Reboot the computer.

Note:

Type EXIT without any parameters to exit WinOne. The current state is automatically saved when exiting WinOne.

Command HELP

Function:

Display help information for a WinOne command.

Syntax:

HELP [command] [/L]

command	Specifies the command or topic to display.
/L	Use the WOIO.HLP file instead of the default WIN_ONE.HLP file.

Note:

Type HELP without any parameters to display the Contents for the WIN_ONE.HLP file.

Command LABEL

Function:

Creates, changes, or deletes the volume label of a disk.

Syntax:

LABEL [drive:][label]

Note:

Use the following syntax to delete a volume label :-

LABEL [drive:]

Command MEM

Function:

Display free memory that is available in the system heap. Also displays, as a percentage, the free system resources for both the GDI and USER heaps.

Syntax:

MEM

Command MKDIR or MD

Function:

Creates a directory.

Syntax:

MKDIR [drive:]path
MD [drive:]path

Command PATH

Function:

Displays or sets the search path environent variable for executable files.

Syntax:

PATH [[drive:]path[;...]]

Note:

Type PATH ; to clear all search-path settings and direct WinOne to search only in the current directory.

Type PATH without any parameters to display the current path.

Command PROMPT

Function:

Changes the command prompt.

Syntax:

PROMPT [text]

text Specifies a new command prompt.

Prompt can be made up of normal characters and the following special codes :-

$B | (pipe)

$C	Current command line number
$D	Current date
$E	Escape code (ASCII code 27), See ANSI Graphics support
$G	> (greater-than sign)
$H	Backspace (erases previous character)
$L	< (less-than sign)
$N	Current drive
$P	Current drive and path
$Q	= (equal sign)
$T	Current time
$V	DOS version number
$$	$ (dollar sign)
$_	Carriage return and linefeed

Note:

Type PROMPT without any parameters to reset the prompt to the default setting.

Command RENAME or REN

Function:

Renames a file or files.

Syntax:

RENAME [drive:][path]filename1 filename2
REN [drive:][path]filename1 filename2

filename1	Specifies the source file(s).
filename2	Specifies the new file name(s).

Note:

A new drive or path cannot be specified for the destination file (filename2).

Command REN can be used to rename directories. When renaming directories Wildcard characters can not be used.

Command RMDIR or RD

Function:

Removes (deletes) a directory.

Syntax:

RMDIR [drive:]path [/S]
RD [drive:]path [/S]

/S	Process sub-directories.

Command SET

Function:

Displays, set, or removes environment variables.

Syntax:

SET [variable=[chars]]

variable	Environment variable name.
chars	A series of characters to assign to the variable.

Note:

Type SET with no parameters to display a list of all environment variables.

When parameter chars is not specified then the environment variable will be removed from the environment space,

Also see Batch Programs.

Command TIME

Function:

Displays or sets the system time.

Syntax:

TIME [time]

time	hh:mm:ss

Note:

Type TIME with no parameters to display the current time setting.

Command TREE

Function:

Graphically displays the directory structure.

Syntax:

TREE [drive:][path]

Command TYPE

Function:

Displays the contents of a text file.

Syntax:

TYPE [drive:][path]filename [/tabstop]

/tabstop	Number between 1 to 8 inclusive. Specifies the number of space characters to use to expand tabstop characters.

Note:

The default number of space characters used to expand tabstop characters is 8.

A file can contains ANSI escape sequences, see ANSI Graphics support.

Command VER

Function:
Display the version numbers for DOS, Windows and WinOne.

Syntax:
VER

Command VERIFY

Function:
Tells WinOne whether to verify that your files are written correctly to a disk.

Syntax:
VERIFY [ON | OFF]

Note:
Type VERIFY without any parameters to display the current setting.

Command VOL

Function:
Display the disk volume label.

Syntax:
VOL

Extra Commands

Command ACS

Function:
Display the full ASCII Character Set.

Syntax:
ACS

Command ARCH

Function:

Displays files inside most popular archive formats, including ZIP, LZH, ARJ, ARC formats.

Syntax:

ARCH [[drive:][path]filename] [/V]

/V Verbose display.

Note:

Type ARCH with no parameters, to display files inside any archive format, in the current directory.

Verbose mode will display the filename, original file size, compressed file size, percentage, date and time for each file inside the archive.

Command CDD

Function:

Changes the current directory and disk.

Syntax:

CDD [drive:][path]

Note:

Also see command CD and GO.

Command DECODE

Function:

Decodes a file given the correct password.

Syntax:

DECODE [drive:][path]filename password

filename Specifies the file(s) to decode.

Note:

Also see command ENCODE.

Command DELBUT

Function:

Deletes all files except the files specified.

Syntax:

DELBUT [drive:][path]filename1 [filename2] [...] [/P]

[drive][path] Specifies the drive and path where the file(s) can be found.
filename1, ... Specifies the file(s) NOT to delete.
/P Prompts for confirmation before deleting each file.

Note:

Also see command DEL.

Example on using DELBUT.

DELBUT Example

To delete all files except the files that have a file extension of .EXE or .COM, in the current directory, enter at the prompt :-

DELBUT *.EXE *.COM

Command DESCRIBE

Function:

Add, modify or delete a file or directory description of up to 60 characters.

Syntax

DESCRIBE [drive:][path]filename [["]description["]]

[drive:][path]filename Specifies the file name or the directory name to be described.
description Specifies a sequence of characters, up to 60 characters.

Note:

When the parameter description is not specified then the file or directory description will be deleted. The file or directory itself will not be deleted.

To include special characters in parameter discription, enclose the discription in a string, for example :-

DESCRIBE WIN_ONE.TXT ">>> Read Me First <<<"

Wildcard characters can not be used with parameter filename.

File and directory descriptions are stored in a hidden file called DESCRIPT.ION, and can be displayed by using the DIR command. Descriptions are automatically maintained when using the commands COPY, DEL, DELBUT, MOVE, RENAME, RMDIR, SCOPY and SMOVE.

Command DISK

Function:

Display bytes used for all sub-directories (including nested sub-directories), of the specified directory.

Syntax:

DISK [[drive:]path]

Note:

Type DISK without any parameters to display the number of bytes used for all sub-directories and nested sub-directories, contained in the current directory.

Command DOS

Function:

Shell to DOS.

Syntax:

DOS

Command DUMP

Function:

Displays the contents of any file.

Syntax:

DUMP [drive:][path]filename [location] [/D][/H][/A][/Z]

filename	Specifies the file to display.
location	Specifies the starting location in the file.
/D	Decimal output.
/H	Hexadecimal output.
/A	Display characters only.
/Z	Display locations starting from zero.

Colours:

The output is colour coded to make it easier to recognise the different character types :-

Magenta	Letter characters
Cyan	Punctuation characters
Yellow	Dot character
Green	Number characters
Red	Null (zero) character
Blue	Carriage return or Linefeed characters
White	Unprintable characters

Note:

The default output is in hexadecimal values.

Parameter location can be specified in either decimal or hexadecimal values. Hexadecimal values must be preceded with the characters '0x'. For example, 255 in hexadecimal is 0xFF.

Command ENCODE

Function:

Encodes a file given a password.

Syntax:

ENCODE [drive:][path]filename password

filename Specifies the file(s) to encode.

Note:

The original file contents are overwrite and then deleted.

Also see command DECODE.

Command FIND

Function:

Search for a text string in the specified files.

Syntax:

FIND [drive:][path]filename "textstring" [/S] [/M]

[drive:][path]filename Specifies the text file(s) to search.
textstring Specifies the text string to find.
/M Match case.
/S Process sub-directories.

Note:

Command FIND uses the fast Boyer-Moore Algorithm.

Command FIND is not compatible with the DOS command FIND.

Unlike the DOS command FIND, Wildcards are allowed for parameter filename.

Also see macro example 4.

Command GO

Function:

Changes the current directory to another sub-directory.

Syntax:

GO [drive:]path

path The target sub-directory to change to.

Note:

This command searches the directory structure to find the first matching sub-directory to change to.

Type GO \path to start searching from the root directory.

Also see command CD and CDD.

Example on using GO.

GO Example

Assume you are in the root directory of D: with the following prompt :-

D:\>

To move to the directory D:\WINDOWS\SYSTEM, enter at the WinOne prompt :-

GO SYSTEM

The NEW prompt will be :-

D:\WINDOWS\SYSTEM>

Command HISTORY or HIS

Function:

Displays the history buffer, which consists of commands entered at the prompt.

Syntax:

HISTORY
HIS

Command MACRO

Function:

Provides command line macro's.

Syntax:

MACRO [macroname=text]

or

MACRO [/D macroname]

or

MACRO [ON | OFF]

macroname	Specifies the name of the new command.
text	Specifies the commands to be executed.
/D	Delete a macro.
ON	Displays expanded macro's.
OFF	Does not display expanded macro's.

The are some special character combinations that have a special meaning and that can be included in parameter text :-

$$ - Dollar sign.

$1 to $9	- Replaceable parameters similar to %1 to %9 parameters used in batch programs.
$*	- Replace with the whole command tail.
$B	- Bar character. Pipe commands.
$G	- Greater than sign character. Redirect output.
$L	- Less than sign character. Redirect input.
$T	- Command separator.

Note:

Type MACRO without any parameters to display a list of all the macro's defined.

Macro's can have the same name as an original command, to execute the original command, type a space before the command is entered.

Macro's are saved as they are created.

Only the first 12 characters in parameter macroname are significant., when executing a macro at the WinOne prompt. However, all the characters in parameter macroname are significant when deleting a macro.

Examples on using MACRO.

MACRO Examples

Example 1:

To create a command LS, which will display a sorted directory listing in lower case letters and in wide format, enter at the WinOne prompt :-

MACRO LS=DIR /OWL $*

To display all .EXE files, enter at the WinOne prompt :-

LS *.EXE

Example 2:

To alter the DIR command to display a sorted directory listing, enter at the WinOne prompt :-

MACRO DIR=DIR /O $*

Example 3:

To create a new command to duplicate a floppy diskette, enter at the WinOne prompt :-

MACRO DUP=TOFILE $1 C:\TMP.DSK$TTODISK $1 C:\TMP.DSK

To duplicate a floppy diskette in A:, enter at the WinOne prompt :-

DUP A:

Example 4:

To create a command to locate a file by its file description only, enter at the WinOne prompt :-

MACRO WHEREIS=FIND *.ION "$*" /S

To locate all the SPX modules, enter at the WinOne Prompt :-

WHEREIS SCREEN SAVER MODULE

Example 5:
To create a command to execute an original internal DOS command, enter at the WinOne prompt :-

MACRO ORIG=C:\COMMAND.COM /C $*

To redirect the DOS DIR command into a file called TEMP.TXT, enter at the WinOne prompt :-

ORIG DIR > TEMP.TXT

Example 6:
To delete the macro ORIG, enter at the WinOne prompt :-

MACRO /D ORIG

Command MORE

Function:
Sets more on or off.

Syntax:
MORE [ON|OFF]

Note:
When more is set on, -- MORE -- will be displayed after each screen full. Press any key to display the next screen.

More always defaults to ON when WinOne is executed.

Command MOVE

Function:
Moves one or more files to another location.

Syntax:
MOVE [/A|/B] source [destination] [/V][/Q]

source	Specifies the file(s) to be copied.
destination	Specifies the directory and/or filename for the new file(s).
/A	Ignored.
/B	Ignored.
/Q	Quiet Mode. Only Error Messages are displayed.
/V	Verifies that new files are written correctly.

Note:
All files are opened in binary mode.

The current directory is used when the parameter destination is not specified.

Also see the commands COPY, SCOPY and SMOVE.

Command SCOPY

Function:

Safely copies one or more files to another location. Will display a warning when a file that already exists is about to be over-writen.

Syntax:

SCOPY [/A | /B] source [destination] [/V]

source	Specifies the file(s) to be copied.
destination	Specifies the directory and/or filename for the new file(s).
/A	Ignored.
/B	Ignored.
/V	Verifies that new files are written correctly.

Note:

All files are opened in binary mode.

The current directory is used when the parameter destination is not specified.

SCOPY does not support appending of files.

When the destination file exists then the user is prompted with the message **Overwrite (Y/N/A/S) ?**. Where entering :-

Y	Yes. Overwrite the file.
N	No. Do not overwrite the file. A new filename is then requested.
A	Always overwrite the files. No further warnings are displayed.
S	Skip this file only.

Also see the commands COPY, MOVE and SMOVE.

Command SMOVE

Function:

Safely moves one or more files to another location. Will display a warning when a file that already exists is about to be over-writen.

Syntax:

SMOVE [/A|/B] source [destination] [/V]

source	Specifies the file(s) to be copied.
destination	Specifies the directory and/or filename for the new file(s).
/A	Ignored.
/B	Ignored.
/V	Verifies that new files are written correctly.

Note:

All files are opened in binary mode.

The current directory is used when the parameter destination is not specified.

When the destination file exists then the user is prompted with the message **Overwrite (Y/N/A/S) ?**. Where entering :-

Y	Yes. Overwrite the file.
N	No. Do not overwrite the file. A new filename is then requested.
A	Always overwrite the files. No further warnings are displayed.
S	Skip this file only.

Also see the commands MOVE, COPY and SCOPY.

Command TODISK

Function:

Copies a TOFILE source image file to a floppy disk in a single pass.

Syntax:

TODISK target [drive:][path]filename [/V]

target	Specifies the target floppy drive to write to. Must be either A: or B:
filename	Specifies the name of the TOFILE source image file.
/V	Set verify off.

Note:

TODISK supports all 5.25" and 3.5" DOS floppy disk formats.

The target floppy disk must be pre-formatted. TODISK does not format disks.

When a image file does not contain the same number of sectors as the target floppy disk, then provided the image is smaller then the target floppy disk, TODISK will be able to copy the image back onto the disk. For example, a 1.2M 5.25" floppy disk image file can be copied onto a 1.44M 3.5" floppy disk. However, the resulting disk will be able to access 1.2M of disk space only.

Also see command TOFILE and macro example 3.

Command TOFILE

Function:

Creates an exact , sector by sector copy of a floppy disk to a single image file in a single pass.

Syntax:

TOFILE source [drive:][path]filename

source	Specifies the source floppy drive to read from. Must be either A: or B:
filename	Specifies the name of the image file to copy the source floppy disk to.

Note:

TOFILE supports all 5.25" and 3.5" DOS floppy disk formats.

Also see command TODISK and macro example 3.

Command WHERE

Function:

Locates one or more files.

Syntax:

WHERE [drive:][path]filename

drive	Specifies the drive to search.
path	Specifies the directory to start searching in.
filename	Specifies the file or files to be found.

Note:

Type WHERE filename, without a drive or path to search the current drive and all non-removable drives on the hard disk.

Examples on using WHERE.

WHERE Examples

Example 1:

To search all drives on the hard disk for all files that have the extension .DOC, enter at the WinOne prompt :-

WHERE *.DOC

Example 2:

To search for files that have the extension .TXT on A: only, starting from the root directory, enter at the WinOne prompt :-

WHERE A:*.TXT

Window Commands

Command ABOUT

Function:

Displays the About window. Also allows the user to register this version of WinOne.

Syntax:

ABOUT

Registration:

Press the Register button in the About window to display the following Registration window :-

Enter the requested information to register WinOne.

Note:
Also see the Shareware Information section for more information on how to obtain a registration number and for the benefits of registering WinOne.

Command DRAG

Function:

Drag one or more files to another program.

Syntax:

DRAG [drive:][path]filename

filename Specifies the file(s) to drag.

Note:

After entering the DRAG command, click the left mouse button in the main WinOne window to pick up the specified file(s), then while holding down the left mouse button, drag the mouse cursor into the window or icon in which to drop the file(s) and release the left mouse button.

Not all programs accept files that are dragged over there window or icon, the cursor is displayed when a window can not accept the file drop and or {bmc curmulti.bmp} cursor is displayed when a window can accept the file drop.

Also see File Drag and Drop.

Command EXT or EXTENSION

Function:

Set filename associations for file extensions.

Syntax:

EXTENSION [extension=association]
EXT [extension=association]

or

EXTENSION [/D extension]
EXT [/D extension]

extension	Specifies the filename extension.
association	Specifies the program name and parameters.
/D	Delete an extension.

Note:

Do NOT include a dot in the parameter extension.

Type EXT without any parameters to display a list of all existing associations.

When an association is established for a file extension, then file names (which MUST including the extension) can be entered at the prompt without the need to specify the program name.

Multiple file extension associations are created and deleted in the same way as single file extension associations. When a single association already exists then WinOne will prompt the user with the message **Multiple XXX extension (Y/N)?**, in which case typing 'Y' will append the association, or typing 'N' will over write the existing association. Also when deleting Multiple associations, WinOne will confirm each association before deleting it.

File extension associations are saved as they are created.

Also see File Extension Association.

Examples on using EXTENSION.

EXTENSION Examples

Example 1:

To associate all file names with an extension of TXT to the program NOTEPAD.EXE, enter at the WinOne prompt :-

EXT TXT=NOTEPAD.EXE ^.TXT

The caret character (ie. ' ^ '), will be replaced with the filename excluding the extension, hence the .TXT extension must be specified.

To execute the program NOTEPAD and pass the file REPORT.TXT to the program, enter at the WinOne prompt :-

REPORT.TXT

Example 2:
Assume that the drive and directory for the program VPIC.EXE is located in D:\PIC. Then to create an association to view GIF files, enter at the WinOne prompt :-

EXT GIF=D:\PIC\VPIC.EXE ^ .GIF

To view GIF files, located in D:\CARS, enter at the WinOne prompt :-

D:\CARS*.GIF

Example 3:
To delete the file extension association for .TXT files, enter at the WinOne prompt :-

EXT /D TXT

Special Notes on associations:
Wildcards can be entered at the WinOne prompt, but the filename extension must not contain Wildcards, since the filename extension is used to determine which association should be used.

When an association is determined, then WinOne will try to locate the program. WinOne will search the current directory first, and if it is not found there then the PATH environment variable is searched.

Command GROUP

Function:
Insert program files into Program Manager groups.

Syntax:
GROUP [["groupname"] filename [/D][/S]]

or

GROUP ["groupname" /D]

or

GROUP [ON | OFF]

groupname	Program Manager group name string.
filename	Program file(s) to add to the group.
/D	Delete a group.
/S	Process sub-directories.
ON | OFF	Enable or disable custom Program Manager group icons.

program files:
Only valid program files can be inserted into the Program Manager groups, and all other file types are ignored. A valid program file must have an extension of either .EXE, .COM, .BAT, .PIF or any filename that has a File Extension Association.

error message:
The command GROUP uses Dynamic Data Exchange to communicate with the Program Manager. A request is sent to the Program Manager, and the Program Manager returns either a positive or negative acknowledgment, depending on whether the request is carried out or not.

When a request fails then the error message **Program Manager request failed** is displayed. The most common reasons for this error occurring is :-

1. WinOne cannot link to the Program Manager.

2. an attempt to delete a Program Manager group that does not exist.

3. an attempt to insert a program file into a Program Manager group that already contains 50 program files.

Note:

Type GROUP without any parameters to display all Progman groups and files within these groups.

Also see Progman Manager Group Icons.

Examples on using GROUP.

GROUP Example

Example 1:

Assume the current directory is D:\WIN_ONE. To insert all .EXE files into a Program Manager group **Windows DOS Shell**, enter at the WinOne prompt :-

GROUP "Windows DOS Shell" *.EXE

Example 2:

To insert all program files into the Program Manager group **Windows App**, that is, files contained in the directory D:\WINDOWS, including all sub-directories, enter at the WinOne prompt :-

GROUP "Windows App" D:\WINDOWS*.* /S

Example 3:

To delete the Program Manager group **Windows App**, and insert the program SYSEDIT.EXE into the group **Windows App**, enter at the WinOne prompt :-

GROUP "Windows App" D:\WINDOWS\SYSEDIT.EXE /DS

Example 4:

To delete the Program Manager group **Windows App**, enter at the WinOne prompt :-

GROUP "Windows App" /D

Command LOCK

Function:

Locks the terminal until the correct password is entered.

Syntax:

LOCK [ON | OFF]

ON Enable the screen saver.
OFF Disable the screen saver.

Note:

Type LOCK with no parameters to activate the screen saver. A random enabled SPX module will be selected and displayed. Press any key or move the mouse to bring up the password window. Enter the correct password to return to Windows.

When the screen saver is enabled or disabled on the command line, the environment variable LOCK is set to the previous ON or OFF setting. For example :-

```
@ECHO OFF
LOCK OFF
...
...
...
LOCK %LOCK%
```

Also see Screen Saver and Terminal Lock to enable or disable the password.

Command MARK

Function:

Mark an application meant for an earlier version of Windows to run under Windows 3.1.

Syntax:

MARK filename

filename Valid executable file with extension .EXE

Note:

Wildcard characters are not allowed.

When an application is successfully marked then Windows will not warn the user, that a application is meant for an earlier version.

Command MODULES

Function:

Display the currently loaded modules.

Syntax:

MODULES

Note:

The MODULES command display the module handle, name, reference count and path for each module that is loaded into memory.

Also see command TASKS.

Command PRINT

Function:

Print a text file via the Printer Manager.

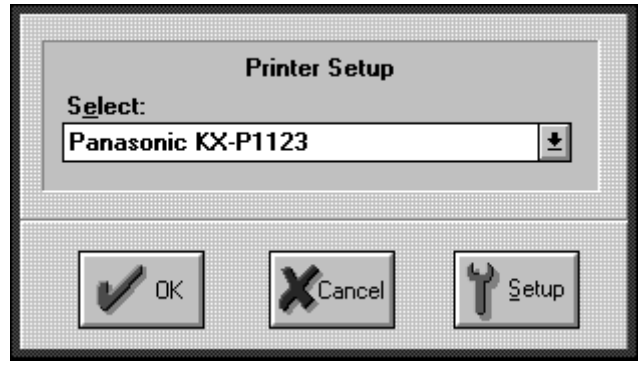
Syntax:

PRINT [[drive:][path]filename] [/tabstop] [/S] [/N] [/L]

filename	Specifies the file to print.
/tabstop	Number between 1 to 8 inclusive. Specifies the number of space characters to use to expand tabstop characters.
/S	Small font size.
/N	Normal font size. This is the default font size used to print a file.
/L	Large font size.

Print Setup:

Type PRINT with no parameters to display the Printer Setup window :-



Select the printer to configure and press the Setup button to change that printers configuration. Changes to a printers configuration is reflected system wide.

Note:

Wildcards are not allowed for parameter filename.

Command RAISE

Function:

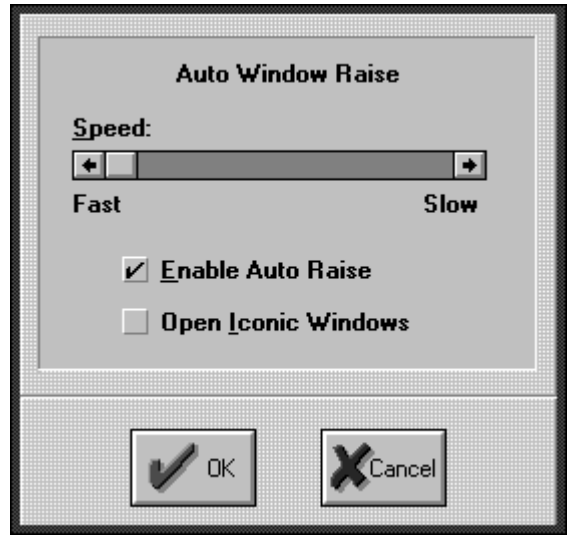
Displays the Auto Window Raise window.

Syntax:

RAISE

Description:

The Auto Window Raise window displays the speed bar, the Enable Auto-Raise check box, and the Open Iconic Windows check box :-



When Auto-Raise is enabled, use the mouse and place the cursor anywhere inside the window to be raised, and when the time delay expires the window will be raised automatically, there is no need to manually raise the window by pressing the left mouse button. The speed bar sets the time delay that must expiry before a window is raised.

Command SENDKEYS

Function:

Send a number of key strokes to the specified window.

Syntax:

SENDKEYS "who" "keys" [/C]

who	A string that specifies the window to send the keys to. This can be either a window caption or a window class. By default who is a window caption, use the /C switch to specify a window class.
keys	A string containing a sequence of characters which specifies the key strokes to be sent.
[/C]	Specifies that parameter who refers to a window class.

Specifying the key strokes

The string containing the key strokes can include any print-able ASCII characters, that is, characters in the ASCII range of 32 to 126 inclusive. However, there are a number of characters that have a special meaning, these include :-

Short Hand Character

~	Press the ENTER key.

Change State Characters. These characters change the state of the next key pressed or a Sub Group of key presses.

+	Hold down the SHIFT key.
^	Hold down the CONTROL key.
%	Hold down the ALT key.

Group Characters.

{ }	Used to specify a Special Key and/or give a key a Repeat Count.
( )	Used to group a number of keys togetther, that is, to specify a Sub Group.

Note:

The maximum number of characters that can be sent is 128.

Examples for parameter keys.

Special Keys

There are a number of Special Keys that cannot be specified with a single character, instead, they are specified by enclosing the name of the key in braces. For example, the Caps Lock key is specified by "{CAPLOCK}".

Valid Special Key names include :-

CAPSLOCK	Caps Lock
NUMLOCK	Num Lock
SCROLLOCK	Scroll Lock
ESCAPE or ESC	Escape
ENTER	Enter
PRTSC	Print Screen
TAB	Tab
BREAK	Break
BACKSPACE or BS	Back Space
DELETE or DEL	Delete
INSERT	Insert
LEFT	Left Arrow
RIGHT	Right Arrow
UP	Up Arrow
DOWN	Down Arrow
PGUP	Page Up
PGDN	Page Down
HOME	Home
END	End
F1 to F12	Function keys F1 to F12

Similarly, the characters ~+^%(){} each have some special purpose or meaning, and to specify the equivalent key, the character must be enclosed in braces. For example, the plus key is specified by "{+}" and not "+".

Repeat Count

Any key except the SHIFT, CONTROL and ALT keys can be repeated or pressed a number of times, by enclosing the key and the repeat count, separated by a space character, in braces. For example, moving the cursor left 10 positions (ie. pressing the left arrow key 10 times), is specified by "{LEFT 10}".

Sub Groups

To change the state of a number of keys, that is, hold down either the SHIFT, CONTROL or ALT key, while pressing a number of other keys, then the other keys must be enclosed in brackets. For example, to move the cursor back three words, that is, hold down the CONTROL key, while pressing the left arrow key three times, is specified by "^({LEFT}{LEFT}{LEFT})".

Examples

Parameter keys	Valid	Output
"abcDEF"	OK	abcDEF
"abcdef{BACKSPACE}"	OK	abcde
"abcdef{BACKSPACE 3}"	OK	abc
"BACKSPACE"	OK	BACKSPACE
"abcghi{LEFT 3}DEF"	OK	abcDEFghi
"{a 5}"	OK	aaaaa
"+{a 5}"	OK	AAAAA
"abc+(def)"	OK	abcDEF
"+abc+def"	OK	AbcDef
"{{}"	OK	{
"{+}"	OK	+
"{+ 5}"	OK	++++
"(abc)"	OK	abc
"{(}abc{)}"	OK	(abc)
"+{+a 5}"	Error - '+a' is not a valid key	
"{abc}"	Error - 'abc' is not a valid key	
"{DELETE"	Error - missing brace	

Command TASKS

Function:

Display a list of the current tasks or performs a specified action on a task.

Syntax:

TASKS [[hwnd] state]

or

TASKS [["caption"] state]

hwnd	Window handle.
caption	Window Caption string.
state	Specifies the action :-

CLOSE	terminate a program
MIN	display window as icon
MAX	display maximised window
SHOW	display a previously hidden window
HIDE	hide window
RESTORE	display window in it original size and position

Note:

Type TASKS without an parameters to display a list of window handles, captions and class names.

When the parameter hwnd or caption is not specified then the action is performed on WinOne itself.

Also see command MODULES.

Command VIEWICON

Function:

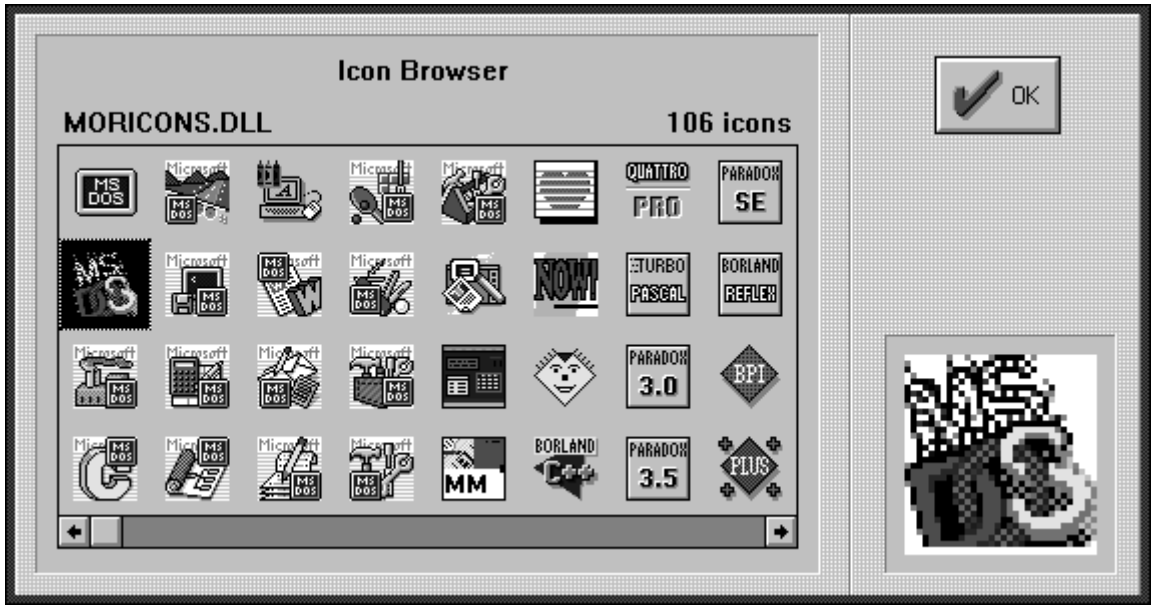
Displays icons in .EXE, .DLL and .ICO files.

Syntax:

VIEWICON [[[drive:]path]filename]

Description:

The VIEWICON command will display a window containing all icons in the specified file(s), as follows :-



To determine the filename for an icon, position the mouse cursor over the icon and press the left mouse button.

Command WALLPAPER

Function:

Set or remove the Desktop Wallpaper.

Syntax:

WALLPAPER [[drive:][path]]filename] [/C][/T]

filename	Specifies the bitmap file name.
/C	Centre the bitmap.

/T Title the bitmap.

Note:

Type WALLPAPER without any parameters to remove the current Wallpaper from the Desktop.

Also see Random Desktop Wallpaper.

External Commands

External Commands are programs that use the main WinOne window for there screen output. Since these commands are simply programs, they require the PATH environment variable to be set in AUTOEXEC.BAT, to include the WinOne directory, so that WinOne can locate them.

Command AUTOGEN (External)

For more information on command AUTOGEN, see the WOIO.HLP help file, which is located in the sub-directory WOIO.

Command ARGS (External)

For more information on command ARGS, see the WOIO.HLP help file, which is located in the sub-directory WOIO.

Command BASE (External)

Function:

Convert between binary, octal, decimal and hexadecimal numbers.

Syntax:

BASE number [/V]

number Specifies the number to convert. The last character in the number specifies the base and can include :-

- b - binary number (eg. 101010b).
- o - octal number (eg. 377o).
- d - decimal number (eg. 512d), This base is the default and the 'd' is optional.
- h - hexadecimal number (eg. 74CFh).

/V Show version information.

Note:

Command base displays the equivalent binary, octal, decimal and hexadecimal numbers.

Command CALEND (External)

Function:

Display a monthly or yearly calendar.

Syntax:

CALEND [year] [month] [/V]

year	yyyy
month	mm
/V	Show version information.

Note:

The system date is used when a parameter is not specified.

When parameter month is specified, the requested month as well as the previous and following months are displayed.

When only the parameter year is specified then a calendar for the whole year is displayed.

Command CMP (External)

Function:

Compare two files byte by byte.

Syntax:

CMP filename1 filename2 [/V]

filename1, filename2	Specifies the files to compare.
/V	Show version information.

Note:

Parameters filename1 and filename2 can include an optional path for each file name.

The output is separated into three columns. The first column is the decimal offset into the files, for the two bytes that differ. The second column is the hexadecimal value for the byte that differs in the first file (ie. filename1). Similarly, the third column is the hexadecimal value for the byte that differs in the second file (ie. filename2). When both file are identical there is no output.

When comparing two files with each other and the file sizes differ, then CMP will still compare the two files, up until the end of which ever file is the smaller one. The message EOF on file name, is displayed in this case.

Wildcards can not be used for parameters filename1 and filename2.

Also see command DUMP.

Command DECOMP (External)

Function:

Decompress a UNIX compressed file (ie. .Z file).

Syntax:

DECOMP source destination [/V]

source	Specifies the file to decompress.
destination	Specifies the file name of the decompressed file.
/V	Show version information.

Note:

Parameters source and destination can contain an optional path. When the destination file exists, then DECOMP will fail and display an error message.

Wildcards can not be used for parameters source or destination.

Also see command DETAR.

Special Note:

There should be no need to ever tar or compress a file under DOS. There are a number of archive formats that are supported by both UNIX and DOS (eg. ZIP archive format), which should be used instead.

Command DETAR (External)

Function:

Detar a UNIX tape archive (ie. .TAR file).

Syntax:

DETAR [drive:][path]filename[.TAR] [/D] [/V]

filename	Specifies the file to detar.
/D	Display a directory listing of the tar archive.
/V	Show version information.

Note:

A tar file must have a file extension of .TAR, since this is the only way to determine whether or not a file is a tar file.

The directory structure inside the tar file is recreated and the file names inside a tar file are mapped to DOS compatible file names.

When a file already exists, then the user will be prompted whether to over write the existing file and if not, a new file name will be requested. It is quite possible for a tar file to contain more than one file name that will map to a single DOS file name, since under UNIX filenames are case sensitive. For example, Read.Me and read.me will both be mapped to READ.ME under DOS.

Wildcards can not be used for parameter filename.

Also see command DECOMP.

Special Note:

There should be no need to ever tar or compress a file under DOS. There are a number of archive formats that are supported by both UNIX and DOS (eg. ZIP archive format), which should be used instead.

Command DOSUNIX (External)

Function:

Convert DOS text files to Unix text files.

Syntax:

DOSUNIX [drive:][path]filename [/V]

filename	Specifies the file(s) to convert.
/V	Show version information.

Note:

This command will not alter a Unix text file.

Also see command UNIXDOS.

Command DUPLICAT (External)

Function:

Locate all duplicate files on the specified drive(s).

Syntax:

DUPLICAT [drive1 drive2 ...] [/N][/S][/C]

drive1 drive2 ...	Specifies the disk drive(s) to search.
/N	By filename only. This will locate files that have the same name. The files found can differ in size or content.
/S	By size and filename. This is the default, when no switches are specified. This will locate files the have the same name and size. The files found can differ in content.
/C	By size, filename and content. This will locate all files that are exactly the same.

Note:

When no drives are specified DUPLICAT will search all non-removable hard disks.

The /N switch is the fastest and the /C switch is the slowest.

Command FUNC (External)

Function:

Call a Windows API function at run time.

Syntax:

FUNC module function [params] [/I] [/L] [/V]

module	Specifies the module name that contains the function to be called. Can be either a file name (eg. TOOLHELP.DLL) or an already loaded module name (eg. USER).
function	Specifies the name of the function to be called.
params	Specifies the parameters to be past to the function. Valid parameters can include the following types :- 1. 16 bit integer values, which are specified as a sequence of digits (eg. 1024). 2. 32 bit long integer values, which are specified as a sequence of digits, followed by the character 'L' (eg. 1048576L). 3. strings, which are enclosed in double quote-mark characters. 4. null, which specifies a NULL pointer (ie. a value of 0L). 5. buf, which specifies a pointer to an internal buffer, which can be used to store information that some functions need.
/I	Specifies the return value is a 16 bit integer value.
/L	Specifies the return value is a 32 bit long integer value.
/V	Show version information.

Note:

Command FUNC can only call PASCAL type functions and can not call CDEL type functions.

All Windows Dynamic Link Libraries contain functions (ie. exports) that can be called at run time. There are a number of modules that are loaded automatically when Windows starts up, including KERNEL, USER and GDI.

The return value is stored in the environment variable %FUNCA%, as a sequence of digits. Long integer return values are also stored in the this variable, but the value does not contain the trailing 'L' character.

When the parameter buf is used, then the contents of the internal buffer is stored in the environment variable %FUNCB%.

Also see Batch Programs.

Examples on using FUNC.

Warning

Command FUNC should only be used by programmers that have some experience in programming Windows applications, since, due to the nature of the way PASCAL functions are implemented, it is possible to crash Windows, however this will not generally be the case.

CDEL functions have their parameters pushed onto the stack before the function is called and popped of the stack after the function returns. Similarly, PASCAL functions have their parameters pushed onto the stack before the function is called, however, the function that is been called is then responsible for popping the parameters of the stack before the function returns. Therefore, if a PASCAL function is called without the right type or number of parameters, then this will cause a hopefully harmless General Protection fault.

FUNC Examples

There are three example batch programs, that are supplied with WinOne and illustrates the use of the command FUNC :-

ARRANGE.BAT - Which arranges all the iconic windows on the desktop. This batch command performs the same function as clicking on the Arrange Icons button in the Task Manager. This batch command takes no parameters.

```
@echo off

rem É[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[»
rem ˚                                                    ˚
rem ˚                    ARRANGE.BAT                      ˚
rem ˚                                                    ˚
rem ˚ Example  batch  program  that  illustrates  the ˚
rem ˚ use of the external command FUNC.  This program ˚
rem ˚ must be run from the WinOne prompt.              ˚
rem ˚                                                    ˚
rem ˚ Description: Arranges all the iconic windows on ˚
rem ˚              the desktop.                         ˚
rem ˚                                                    ˚
rem ˚                                                    ˚
rem ˚   Written by Lucien Cinc (1993)                   ˚
rem ˚                                                    ˚
rem È[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[¼

func user getdesktopwindow
func user arrangeiconicwindows %funca%
if %funca%==0 goto error
goto end

:error
color red black
echo Unable to arrange icons
echo.
```

:end

CASCADE.BAT - Cascades all the windows on the desktop. This batch command performs the same function as clicking on the Cascade button in the Task Manager. This batch command takes no parameters.

```
@echo off

rem É[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[»
rem ˚                                                    ˚
rem ˚                    CASCADE.BAT                      ˚
rem ˚                                                    ˚
rem ˚ Example  batch  program  that  illustrates  the ˚
rem ˚ use of the external command FUNC.  This program ˚
rem ˚ must be run from the WinOne prompt.              ˚
rem ˚                                                    ˚
rem ˚ Description: Cascade  all  the  windows  on the ˚
rem ˚              desktop.                             ˚
rem ˚                                                    ˚
rem ˚                                                    ˚
rem ˚   Written by Lucien Cinc (1993)                   ˚
```

```
rem ³                                                    ³
rem ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

func user getdesktopwindow
func user cascadechildwindows %funca% 0
if errorlevel 1 goto error
goto end

:error
color red black
echo Unable to cascade windows
echo.

:end
```

TILE.BAT - Tiles all the windows on the desktop. This batch command performs the same function as clicking on the Tile button in the Task Manager. This batch command takes no parameters.

```
@echo off

rem ÉÍÍÍ
```

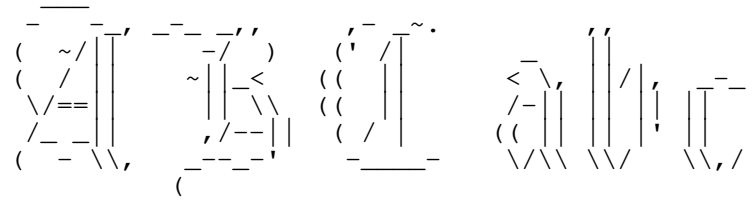
Display small horizontal Gothic style characters.

Syntax:

GOTHIC [msg] [/V]

msg	Sequence of characters.
/V	Show version information.

Sample:



Note:

Also see command SBANNER.

Command LOGO (External)

Function:

Change the Windows Start up logo.

Syntax:

LOGO [drive:][path][filename] [/V]

filename	Specifies the file containing the new logo. This file MUST be a Run Length Encoded 16 colour bitmap, which is no larger than 48K in size, otherwise an appropriate error message is displayed and the logo will NOT be changed.
/V	Show version information.

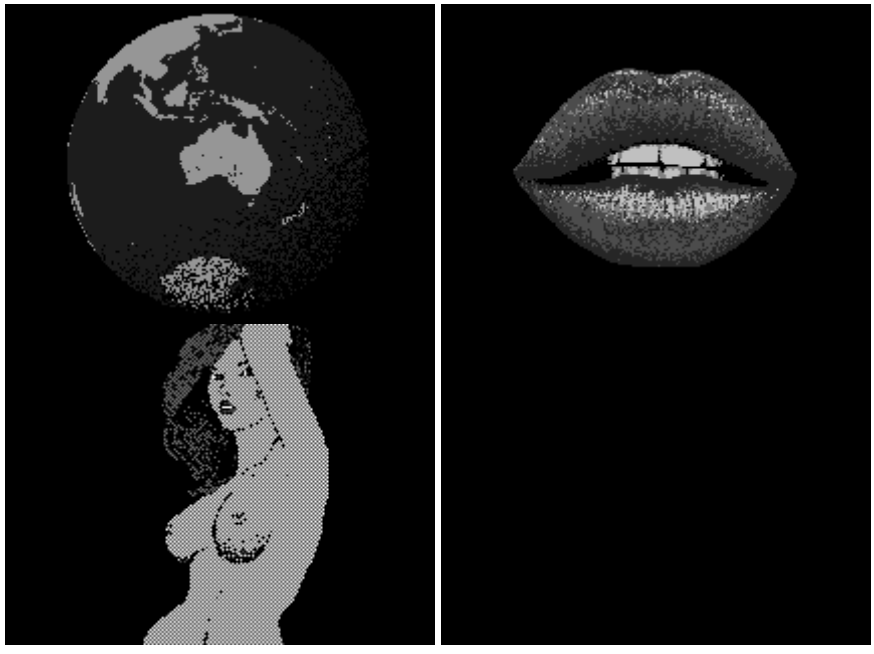
Note:

Type LOGO without any parameters to restore WIN.COM to its original state.

The first time command LOGO is used to change the Windows logo, then a backup copy of WIN.COM is made to the file WINBAK.COM. Should something go wrong, then you will be able to get back into Windows by running WINBAK.COM at the DOS prompt.

The following images are supplied with WinOne in the files :-

1. GLOBE.RLE
2. LIPS.RLE
3. SEXY.RLE
4. BLACK.RLE



Acknowledgments

The above images have been down loaded from various BBS's and are drawn by a number of unknown, but very talented artists.

Run Length Encoded (RLE) Bitmaps

Run Length Encoding refers to the compression method that is used to compress a bitmap. Typically, a bitmap compressed using this method will have a file extension of .RLE, but this is not generally the case, so command LOGO checks the bitmap internally to verify that is does meet the specified requirements before changing the start up logo.

Due to the nature of RLE bitmaps, that is, it replaces long runs of the same colour, with the number of times the colour occurs, it is possible to compress a bitmap and end up with a file that is larger than the un-compressed bitmap. Typically, a bitmap that has relatively large areas of the same colour, for example, a black background colour, (ie. as with the examples supplied with WinOne), the bitmap should compress to no greater than 48K.

The are many public domain Windows Graphics Packages that support RLE bitmaps, which can be used to convert a bitmap to an RLE bitmap.

Command MERGE (External)

Function:

Merge two files together.

Syntax:

MERGE first second destination [/V]

first	Specifies the first file.
second	Specifies the second file.
destination	Specifies the destination file.

/V Show version information.

Note:

The file first is copied to the file destination, then the file second is appended to the file destination.

Also see command SPLIT.

Command SBANNER (External)

Function:

Display small horizontal characters.

Syntax:

SBANNER [msg] [/V]

msg Sequence of characters.
/V Show version information.

Sample:

```
   ##   ######    ####                    ###
  ####    ##  ##  ##  ##                    ##
 ##  ##   ##  ## ##                 ####    #####    ####
 ##  ##   #####  ##                    ##   ##  ##  ##  ##
 ######   ##  ## ##                 #####   ##  ##  ##
 ##  ##   ##  ##  ##  ##           ##  ##   ##  ##  ##  ##
 ##  ## ######    ####             ### ## ## ###    ####
```

Note:

Also see command GOTHIC.

Command SHRED (External)

Function:

Delete and destroy a single file.

Syntax:

SHRED [drive:][path]filename [/V]

filename Specifies the file to shred.
/V Show version information.

Note:

When a file is shredded the contents of the file are over-written, so that, it can NOT be undeleted.

Wildcards are not allow for parameter filename.

Also see command DEL.

Command SPLIT (External)

Function:

Split a file into two files.

Syntax:

SPLIT [drive:][path]filename size [/V]

filename	Specifies the file to split.
size	Specifies the size in bytes for the first file. Size bytes from the file filename will be placed into the first file and the remaining bytes will be placed into the second file.
/V	Show version information.

Note:

The resulting two files have an extension of .1 for the first file and .2 for the second file.

Wildcards can not be used for parameter filename.

Also see command MERGE.

Command STRINGS (External)

Function:

Display all the strings inside a binary file.

Syntax:

STRINGS [nchars] [drive:][path]filename [/V]

nchars	Specify the minimum number of characters in a string. The default is 5.
filename	Specifies the file to search.
/V	Show version information.

Note:

Wildcards are not allowed for parameter filename.

Valid strings consist for any printable characters, this does not include any control characters.

Command TITLE (External)

Function:

Display the title string contained inside a Windows executable.

Syntax:

TITLE [[drive][path]filename] [/V]

filename	Specifies the file(s) to search.
/V	Show version information.

Note:

When no filename is specified then the title strings for all executable's in the current directory are displayed.

Windows executable's do not need to have a .EXE file extension, for example .SPX, .FON, .DLL etc, all contain title strings.

Command TOUCH (External)

Function:

Touch the date and time for a file(s).

Syntax:

TOUCH [drive:][path]filename [date] [time] [/V]

filename	Specifies the file(s) to touch.
date	dd-mm-yyyy
time	hh:mm:ss
/V	Show version information.

Note:

The system date and/or time is used when parameters date and/or time are not specified.

When parameter filename specifies a file that does not exist, then that file is created and then touched.

Command TOUCH can not be used to change the date or time for a directory.

Command UNIXDOS (External)

Function:

Convert Unix text files to DOS text files.

Syntax:

UNIXDOS [drive:][path]filename [/V]

filename	Specifies the file(s) to convert.
/V	Show version information.

Note:

This command will not alter a DOS text file.

Also see command DOSUNIX.

Standard Batch Commands

Command CALL

Function:

Run a second batch program, then returns to the first batch program.

Syntax:

CALL [drive:][path]filename [batch-parameters]

[drive:][path]filename	Batch program name and location.
batch-parameters	Any command line information that is used by the batch program.

Note:

If you do not need to return to the original batch program, then use the following syntax, without the CALL command :-

[drive:][path]filename [batch-parameters]

The CALL command is used from the command line to run a batch program, the batch program will be executed using DOS and will appear inside a DOS window.

Also see Batch Programs.

Command ECHO

Function:

Display a message or turns echo on or off.

Syntax:

ECHO [message]

or

ECHO [ON|OFF]

message	Sequence of characters to display.
ON	Display commands as they are carried out.
OFF	Do not display commands as they are carried out.

Note:

Type ECHO with no parameters to display whether echo is no or off.

If you wish to display a blank line type ECHO followed by a period :-

ECHO.

Also see Batch Programs.

Tip:

To prevent a single command in your batch program from being displayed, put an at sign in front of the command, for example :-

@ECHO OFF

Command FOR

Function:

Perform a command for each file in the specified set of files.

Syntax:

FOR %variable **IN (** set **) DO** command [parameters]

%variable Specifies a replaceable parameter
(set) Specifies a set of files. Wildcards can be used and file names can be separated by comma's.
command [parameters] Specifies the command and the parameters past to the command, for each file in found in the set.

Note:

When using the FOR command inside a batch program, use %%variable and not %variable.

Also see Batch Programs.

Command GOTO

Function:

Switch to another part of the batch program, and continue executing the program from that point.

Syntax:

GOTO label

label Sequence of characters.

Note:

The label must also appear on its own line elsewhere in the batch program, preceded by a colon, for example :-

```
GOTO end
...
...
...
:end
```

Also see Batch Programs.

Command IF

Function:

Perform conditional processing in a batch program.

Syntax:

IF [NOT] condition command

condition	Is one of the following :-	
	ERRORLEVEL number	Specifies a true condition if the last program returned an exit code greater than or equal to number.
	EXIST filename	Specifies a true condition if the filename exists.
	EXISTWINDOW "text"	Specifies a true condition if the window caption "text" exists.
	EXISTCLASS "text"	Specifies a true condition if the window class "text" exists.
	string1==string2	Specifies a true condition if string1 and string2 are the same.
command	Specifies the command to carry out if the condition is met.	
NOT	Carry out the command only if the condition is false.	

Note:

The parameters string1 and string2 do not need to be enclosed in quote characters.

Also see Batch Programs.

Command PAUSE

Function:

Suspend processing of a batch program and displays a message.

Syntax:

PAUSE [message]

message Sequence of characters.

Note:

Type PAUSE without an parameters to display the message :-

Press any key to continue . . .

Also see Batch Programs.

Command REM

Function:

Allows comments inside a batch file.

Syntax:

REM anything

anything Sequence of characters.

Note:

REM commands are just ignored by WinOne.

Also see Batch Programs.

Command SHIFT

Function:

Change the position of replaceable parameters in a batch program.

Syntax:

SHIFT

Note:

The SHIFT command changes the values of replaceable parameters %0 through to %9, by copying each parameter into the previous one.

When there are more then 10 command line parameters, each will be shifted one at a time into %9.

Also see Batch Programs.

Enhanced Batch Commands

Command ASK

Function:

Ask a yes/no question and set the errorlevel respectively.

Syntax:

ASK [message]

message Sequence of characters.

Note:

Type ASK without any parameters to display the message :-

Continue (Y/N)?

The errorlevel is set to 0 for a yes response, 1 for a no response.

Example on using ASK.

Also see Batch Programs.

ASK Example

Consider the following batch program, which will display a yes/no message, and depending on the user response the program will either continue or end :-

```
ASK Are you sure (Y/N)?
IF ERRORLEVEL  1 GOTO  no
...
```

```
...
...
:no
```

Similarly, the following batch program will display a yes/no message and either loop or end :-

```
:loop
...
...
...
ASK Again (Y/N)?
IF ERRORLEVEL 1 GOTO no
GOTO loop
:no
```

Command BEEP

Function:

Send a beep to the system speaker.

Syntax:

BEEP

Command Box

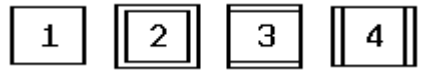
Function:

Display a box in one of four pre-defined formats.

Syntax:

BOX left top right bottom [type]

left top right bottom Screen co-ordinates of box.
type Specifies a number 1 to 4, which represents one of the following box formats :-



Note:

Co-ordinates start from 0. For example, the first character on the screen is at co-ordinate 0, 0.

When parameter type is not specified then format 1 is used.

Also see Batch Programs.

Command CALC

Function:

Perform basic arithmetic calculations in a batch program. The result is stored in an environment variable called %CALC%.

Syntax:

CALC [?] expr

?	Display the result only. Does not set the environment variable.
expr	Sequence of tokens, that can include the following :-

+	Addition
-	Subtraction
*	Multiplication
/ or \	Division
(	Left bracket
)	right bracket
0..9	digits

Note:

Example on using CALC.

Also see Batch Programs.

CALC Example

Example 1:

The CALC command can be used to perform loop incrementing in a batch program, consider the following batch program :-

```
@ECHO off
SET calc=0
:loop
IF %calc% == 10 GOTO stop
ECHO  %calc%
CALC  %calc% + 1
GOTO loop
:stop
```

This program will display the numbers 0 to 9, one to a line.

Example 2:

Consider the following batch program :-

```
@ECHO  off
CLS
SET  text=hello, world
STRLEN  text
CALC  ( 80 -  %strlen% ) / 2
LOCATE %calc% 0
ECHO  %text%
```

This program will clear the screen and display the text 'hello, world', centred on the first line.

Command COLOUR

Function:

Set Colours on or off. Also sets foreground and background colours inside a batch program.

Syntax:

COLOUR [ON | OFF | MONO]

or

COLOUR foreground background

ON	Enable all colours.
OFF	Disable colours. Black text on white background only.
MONO	Disable colours. Gray text on black background only.
foreground	Text foregound colour .
background	Text background colour.

Valid foregound and background colours include :-

WHITE
GRAY
BLACK
RED
GREEN
BLUE
CYAN
MAGENTA
YELLOW

Note:

Also see Batch Programs.

Command DIRS

Function:

Display the directory stack.

Syntax:

DIRS

Note:

The directory stack can hold at most 10 directories. When more than 10 directories are pushed on to the stack then the oldest directories are removed to make room for the new directories.

Use the commands PUSHD and POPD to manipulate the stack.

Also see Batch Programs.

Command END

Function:

End a batch program.

Syntax:

END

Note:

Also see Batch Programs.

Command GETKEY

Function:

Wait for a single keypress from the user. The character entered is stored in an environment variable called %GETKEY%.

Syntax:

GETKEY [message]

message Sequence of characters.

Note:

Also see Batch Programs.

Command GETSTR

Function:

Wait for a sequence of keypresses from the user. The sequence of characters is stored in an environment variable called %GETSTR%.

Syntax:

GETSTR [message]

message Sequence of characters.

Note:

Also see Batch Programs.

Command GOSUB

Function:

Jump to another part of a batch program and continue executing the batch program from that point onwards until a RETURN command is found. Execution of the batch program will then return to the next command following the original GOSUB command.

Syntax:

GOSUB label

label Sequence of characters.

Note:

The label must also appear on its own line elsewhere in the batch program, preceded by a colon, for example :-

```
GOSUB subproc
...
...
:subproc
...
...
RETURN
```

GOSUB's can be nested a maximum of 8 levels.

Also see Batch Programs.

Command LOCATE

Function:

Position the cursor anywhere on the screen.

Syntax:

LOCATE x y

x, y	Specifies the co-ordinates.

Note:

Co-ordinates start from 0. For example, the first character on the screen is at co-ordinate 0, 0.

Also see Batch Programs.

Command LOWER

Function:

Convert a text string to lower case. The converted text string is stored in an environment variable called %LOWER%.

Syntax:

LOWER [text]

text	Sequence of characters.

Note:

Also see Batch Programs.

Command PARSE

Function:

Allows a sentence to be broken into pieces. The pieces are stored in environment variables. %PARSEA% contains the text that was extracted, and %PARSEB% contains the remainder of the text.

Syntax:

PARSE [text]

text	Sequence of characters.

Note:

The PARSE command automatically breaks on commas, quotation marks, periods, exclamation marks, colons, backslashes and frontslashes.

Examples on using PARSE.

Also see Batch Programs.

PARSE Example

Consider the following batch program :-

```
@ECHO  off
SET  text=a,b,c,d,e,f
PARSE  %text%
:loop
IF  "%PARSEB%"==""" GOTO stop
SAY   %PARSEA%
PARSE  %PARSEB%
GOTO  loop
:stop
```

This batch program will display the letters ' abcdef ' on a single line.

Command POPD

Function:

Pop a directory from the directory stack and make this directory the current directory.

Syntax:

POPD [number]

number Number of directories to pop of the stack.

Note:

Also see command PUSHD, DIRS and Batch Programs.

Command PUSHD

Function:

Push the current directory onto the directory stack and change to the specified directory.

Syntax:

PUSHD [[drive:]directory]

directory Specifies the directory to change to.

Note:

When no parameters are specified then the current directory is pushed onto the stack only.

Also see command POPD, DIRS and Batch Programs.

Command RETURN

Function:

Return to the next command following the original GOSUB command.

Syntax:

RETURN

Note:

Also see Batch Programs.

Command SAY

Function:

Display a message. This command with NOT add a carriage return - line feed at the end of the message.

Syntax:

SAY [message]

message Sequence of characters.

Note:

Examples on using SAY.

Also see Batch Programs.

SAY Example

Consider the following batch program :-

```
@ECHO off
SAY hello
SAY  world
```

This example will display 'hello world' on the same line.

Command SLEEP

Function:

Do nothing for a time.

Syntax:

SLEEP [seconds]

seconds Specifies the number of second to wait.

Note:

When no parameter is specified then zero seconds is assumed. Similarly zero is assumed for invalid values of parameter seconds.

Also see Batch Programs.

Command STOP

Function:

Stop processing a batch program and continue processing the batch program that called this one (ie. using the CALL batch command).

Syntax:

STOP

Note:

When there is no batch program to continue processing, then the batch program simple ends.

Also see Batch Programs.

Command STRSIZE

Function:

Determine the length of a string. The length is stored in an environment variable called %STRSIZE%.

Syntax:

STRSIZE [text]

text	Sequence of characters.

Note:

Also see Batch Programs.

Command SUBSTR

Function:

Extract a section of text from a text string. The extracted text string is stored in an environment variable called %SUBSTR%.

Syntax:

SUBSTR pos size [text]

pos	Position to start extracting text from, where the first character is at position 1, the second character is at postion 2, etc...
size	Number of characters to extract.
text	Sequence of characters.

Note:

Parameter pos can also be negative. The last character is at position -1, the second last character is -2, etc...

Examples on using SUBSTR.

Also see Batch Programs.

SUBSTR Example

Example 1:
To extract the first 5 characters from the text string 'abcdefgh', enter at the WinOne prompt :-

SUBSTR 1 5 abcdefgh

The environment variable %SUBSTR% will be set to 'abcde'.

Example 2:
To extract 5 characters from the text string 'abcdefgh', start from the third character, enter at the WinOne prompt :-

SUBSTR 3 5 abcdefgh

The environment variable %SUBSTR% will be set to 'cdefg'.

Example 3:
To extract 5 characters from the end of the text string 'abcdefgh' , enter at the WinOne prompt :-
.

SUBSTR -1 5 abcdefgh

The environment variable %SUBSTR% will be set to 'defgh'.

Example 4:
To extract 5 character from the text string 'abcdefg', starting at the third last character in the string, enter at the WinOne prompt :-

SUBSTR -3 5 abcdefgh

The environment variable %SUBSTR% will be set to 'bcdef'.

Command UPPER

Function:
Convert a text string to upper case. The converted text string is stored in an environment variable called %UPPER%.

.
Syntax:
UPPER [text]

text Sequence of characters.

Note:
Also see Batch Programs.

Index

A
about, 44
acs, 34
arch, 34
args, 54
arrange, 58
arrange.bat, 58
ascii, 34
ask, 68
attrib, 27
autogen, 54
B
bar, 22
base, 54
beep, 69
box, 69
button, 20
buttons, 20
C
calc, 70
calend, 55
calendar, 55
calender, 55
call, 65
cascade, 58
cascade.bat, 58
cd, 27
cdd, 34
chdir, 27
clipboard, 15
cls, 28
cmp, 55
color, 71
colour, 71
comp, 55
compare, 55
compress, 55
copy, 28
cut, 15
D
date, 28
decode, 35
decomp, 55
decompress, 55
del, 29
delbut, 35
describe, 35
Desktop, 14
detar, 56
dir, 29
dirs, 71
discription, 35
disk, 36
dos, 36
dosunix, 57

drag, 14, 44
drop, 14, 44
dump, 36
duplicat, 57
duplicate, 57
E
echo, 65
encode, 37
end, 72
erase, 29
exit, 29
ext, 45
extension, 45
F
find, 37
font, 18
fonts, 18
for, 66
func, 57
G
getkey, 72
getstr, 72
go, 38
gosub, 72
gothic, 60
goto, 66
group, 46
H
help, 30
his, 38
history, 38
I
icon, 53
icons, 53
if, 67
L
label, 30
locate, 73
lock, 48
logo, 60
lower, 73
M
macro, 38
mark, 48
md, 30
mem, 30
memory, 30
merge, 62
mkdir, 30
modules, 48
more, 40
move, 40
P
parse, 73
paste, 15
path, 31
pause, 67
popd, 74

print, 49
progman, 13
prompt, 31
pushd, 74
R
raise, 49
rd, 32
rem, 67
ren, 31
rename, 31
return, 75
rmdir, 32
S
say, 75
sbanner, 62
scopy, 41
sendkey, 50
sendkeys, 50
set, 32
shift, 68
shred, 62
sleep, 76
smove, 41
split, 63
status, 22
stop, 76
stopwatch, 22
string, 63
strings, 63
strsize, 76
substr, 76
syntax, 26
T
tab, 6
tabstop, 6
tar, 56
task, 52
tasks, 52
tile, 58
tile.bat, 58
time, 32
title, 64
todisk, 42
tofile, 42
touch, 64
tree, 33
type, 33
U
unix, 24
unixdos, 64
upper, 77
V
ver, 33
verify, 33
version, 33
viewicon, 53
viewicons, 53
vol, 34

volume, 34
W
wallpaper, 53
where, 43