

Command Post User's Guide

Wilson WindowWare, Inc.

2701 California Ave SW ste 212
Seattle, WA 98116

Orders: (800) 762-8383
Support: (206) 937-9335
Fax: (206) 935-7129

Copyright © 1988-1992 by Morrie Wilson.
All rights reserved.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Wilson WindowWare, Inc. Information in this document is subject to change without notice and does not represent a commitment by Wilson WindowWare, Inc.

The software described herein is furnished under a license agreement. It is against the law to copy this software under any circumstances except as provided by the license agreement.

U.S. Government Restricted Rights

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Contractor/manufacturer is Wilson WindowWare, Inc./2701 California Ave SW /ste 212/Seattle, WA 98116

Trademarks

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.
Windows, Word for Windows, and Excel are trademarks of Microsoft Corporation.

WinBatch, Command Post, and File Commander are trademarks of Wilson WindowWare, Inc.

CONTENTS

INTRODUCTION

[System Requirements](#)
[About This Manual](#)
[Notational Conventions](#)
[Acknowledgements](#)

GETTING STARTED

USING COMMAND POST

[Starting Command Post](#)
[The Command Post Window](#)
[Additional Command Post Windows](#)
[Keyboard and Mouse Shortcuts](#)
[About the Menu Files](#)
[Command-line Parameters](#)
[Replacing the Task Manager](#)
COMMAND POST DEFAULT MENUS

[Control Menu](#)
[File Menu](#)
[Networks](#)
[Preferences](#)
[Dir Menu](#)
[View Menu](#)
[Main Menu](#)

COMMAND POST FUNCTIONS

[Introduction and List of Functions](#)
[DirExist](#)
[DiskHide](#)
[DiskReset](#)
[DiskUpdate](#)
[FileHilite](#)
[IntControl](#)
[MsgTextGet](#)
[OtherDir](#)
[OtherUpdate](#)
[Refresh](#)
[SetDisplay](#)

UTILITIES

[Dialog Editor](#)
[WinInfo](#)

INTRODUCTION

Command Post helps you end some of the drudgery of day-to-day tasks when working in Microsoft Windows.

How many times have you tired of changing drives, moving through two subdirectories, finding an executable file from among 43 other filenames, double clicking on it, selecting **File/Open**, then searching for *another* directory in the listbox because your document happens to be in a different directory than the program!

With Command Post you can *automate* that knowledge of where to move around the disk, which directories to go to, which files you want to work on, what size you prefer your window to be, and so on.

With almost 200 functions and commands, Command Post can:

- Run Windows and DOS programs.
- Send keystrokes directly to applications.
- Rearrange, resize, hide, and close windows.
- Run programs either concurrently or sequentially.
- Display information to the user in various formats.
- Prompt the user for any needed input.
- Present scrollable file and directory lists.
- Copy, move, delete, and rename files.
- Read and write files directly.
- Copy text to and from the Clipboard.
- Perform string and arithmetic operations.
- Make branching decisions based upon numerous factors.

And much, much more.

System Requirements

Command Post requires an IBM PC or compatible with a minimum of 2 megabytes of RAM, running Microsoft Windows version 3.0 or higher.

About This Manual

Command Post is an application which uses our Windows Interface Language (WIL). Please refer to the **WIL Reference Manual** for an introduction to WIL, as well as for complete documentation of the many functions available in WIL (and, therefore, in Command Post).

This User's Guide includes only topics and functions which are exclusive to Command Post or which behave differently in Command Post, as well as additions and changes that have been made since the WIL Reference Manual went to press.

Note: Command Post is a **menu file** based implementation of WIL.

Notational Conventions

Throughout this manual, we use the following conventions to distinguish elements of text:

ALL-CAPS

Used for filenames.

Boldface

Used for important points, programs, function names, keystrokes and parts of syntax that must appear as shown.

`system`

Used for items in menus and dialogs, as they appear to the user.

`Small fixed-width`

Used for WIL sample code.

Italics

Used for emphasis, and to liven up the documentation just a bit.

Acknowledgements

Command Post software developed by Morrie Wilson.

Documentation written by Richard Merit.



Help System Engineering by James Stiles, The Stiles Group, (206) 937-3141 CIS 73240,3131 Internet
73240.3131@compuserve.com

GETTING STARTED

Command Post is easy to install. A diskette with your software on it is part of your Command Post package. You will need to insert it into one of your floppy diskette drives. You will also need to have Windows running and the Program Manager active.

To install Command Post, you need to run the file WSETUP.EXE from your installation diskette. You can do this from whichever floppy drive (a or b) you will use to install Command Post. You can use any Windows shell (Program Manager, File Manager, or any other program with a **File, Run** menu item).

You do this by first selecting **File, Run** from the main menu of the shell. Then you enter either A:\WSETUP or B:\WSETUP into the dialog depending on whether you are running wsetup from your a or your b drive. Hit **enter** and WSETUP will take over from there. WSETUP will copy or create the necessary files in a directory of your choice.

USING COMMAND POST

Starting Command Post

There are several ways you can start Command Post:

1. From Program Manager, by double-clicking on the Command Post icon in the Command Post program group.
2. From File Manager, by changing to the directory where Command Post resides and double-clicking on CMDPOST.EXE.
3. By selecting **Run** from the **File** menu of either Program Manager or File Manager, and typing in "CMDPOST.EXE".
4. Automatically when Windows starts up, by adding "CMDPOST.EXE" to the "RUN=" line in the [Windows] section of your WIN.INI file. Alternatively, in Windows 3.1 or higher, you can copy or move the Command Post icon from the Command Post program group to the StartUp program group in Program Manager (please refer to your Windows manual for more information on this procedure).
5. By making Command Post your Windows shell. Edit the "Shell=" line in the [Boot] section of your SYSTEM.INI file as follows:

Shell=CMDPOST.EXE

Save the file and restart Windows. Command Post will now be your shell.

A special Command Post feature:

If you had been loading any programs automatically upon Windows startup, you will either want to add them to the initialization (AUTOEXEC) section of your CMDUSER.CPM menu file (see the sample CMDUSER.CPM file for an example of this), or else configure Command Post to process the RUN= and LOAD= lines of your WIN.INI file by selecting **Auto Run= and Load= Options** from the Command Post **File** menu, under **Preferences**.

The Command Post Window

At the very top of the Command Post window is the **title bar**. At the left end of the title bar is a little box which is called the **Control menu** (also known as the **System menu**). Below the title bar are the pull-down **menus**, all of which are fully-customizable. Beneath the menus are a series of **drive icons**, one for each available drive on your system, with the icon for the currently-logged drive highlighted. To the right of the drive icons is a text string showing the name of the **current directory**. Below that, and filling up most of the Command Post window, is a list of the subdirectories and files contained in the current directory (by default, hidden and system files are not listed, but this can be changed by selecting **Preferences** from the **File** menu). If there are more files than will fit in the display at one time, then a horizontal **scroll bar** will appear at the bottom of the window, allowing you to scroll through the file display.

Additional Command Post Windows

You can run up to four separate instances of Command Post at any one time. Each copy of Command Post has its own set of menus and its own file display, and (at least by default) has a unique window title. You can have each Command Post window display a different directory, which makes it convenient to perform file management operations. Select **Preferences** from the **File** menu to modify the default Command Post window positions and/or window titles (registered version only).

Keyboard and Mouse Shortcuts

This chart shows the keyboard and mouse shortcuts which can be used to navigate through your directories, and to select and run files. You can use the **File, Print Topic...** menu item on this help program's menu bar to print this material out. You will need the Wingdings font (included with Windows 3.1) to print the icons.

Shortcuts

Select a Drive



Click on drive icon with the left button.



Type Ctrl + desired drive letter.

Open another Cmd Post window on a different drive



Doubleclick on drive icon with the left button.



None.

Select a File



Click on desired filename with left button.



Use arrow keys to place selection box over desired file.



Press first letter of filename to move selection box to next file with that letter.

Select Additional Files



Click on additional filenames with the left button while pressing the Shift key.



Click on additional filenames with the right button.



Drag mouse over a group of additional filenames with the right button pressed.



Press the Shift key while moving selection box over a range of filenames.



Press the Ctrl key while moving selection box over filenames you don't want to select.

Toggle the Highlight on Current File



None



Press the Space Bar.

Move to Top of Dir List



None



Press the Home key.



Press Ctrl-Home if you want the currently-selected file(s) to remain highlighted.

Move to End of Dir List



None



Press the End key.



Press Ctrl-End if you want the currently-selected file(s) to remain highlighted.

Select from Current File to Top of List



None.



Press Shift-Home.

Select from Current File to End of List



None



Press Shift-End.

Select All Files



None



Press the Home key, then press Shift-End.

Select or De-select All Files w/Same Extension as a Certain File



Doubleclick on desired file name with the right button.



None.

Change to a Subdirectory



Doubleclick on desired subdirectory name with the left button.



Place selection box over desired directory and press Enter.

Change to the Parent Directory



Click on desired directory in the current path string (next to the drive icons) with the left button.



Doubleclick on the two dots (..) at the top of the dir list.



Press the Backspace key.



Place selection box over the two dots (..) at the top of the dir list and press Enter.

Run a Program



Doubleclick on desired .EXE filename.



Doubleclick on datafile name if it has an [extensions] association in WIN.INI



Place selection box over desired program or datafile and press Enter.

Run a Program as an Icon



Same as running a program, except press the Shift key while clicking the filename.



Same as running a program, except press Shift+Enter.

Run a Program Maximized



Same as running a program, except press the Ctrl key while clicking the filename.



Same as running a program, except press Ctrl+Enter.

About the Menu Files

All the pull-down menus, with the exception of the Control menu, are completely configurable and are defined in **menu files**, which have an extension of CPM.

Command Post allows you to specify from one to 10 separate menu files. The names of these menu files are specified in the [CmdPost] section of the WWW-PROD.INI file. We include four CPM files, which the setup program places in WWW-PROD.INI as follows:

```
Menu0=CMDPOST1.CPM
Menu1=CMDPOST2.CPM
Menu2=CMDUSER.CPM
Menu3=CMDGROUP.CPM
```

You are free to edit or replace these files with ones of your own (but read the section which follows), or to add additional menu files (i.e., **Menu4** through **Menu9**).

Each of these menu files is independent, and is edited individually (refer to the **WIL Reference Manual** for information on menu file structure).

A single menu file can contain one top-level menu item, or it can contain many such menu items. When the menu files are loaded upon startup, they are appended together to form a single unified menu structure. The reason for having multiple menu files is to make Command Post more modular, and to make it easier to edit the menu files with editors like **Notepad** which cannot handle large files.

The following menu files are shipped with Command Post:

CMDPOST1.CPM **CMDPOST2.CPM**

These two menu files contain the core menu items for file and directory management, configuration of the Command Post display window, execution of common applications, and general housekeeping. We do not recommend that beginning users make major modifications to these files until they are comfortable with the use of the WIL language (as discussed in the **WIL Reference Manual**).

Even advanced users should consider the fact that the default versions of these files get revised as new features are added to the language.

The items contained in these two menu files are described in detail in the section titled **Command Post Default Menus**, below.

CMDUSER.CPM

This menu file contains a number of sample menu items for you to use. Feel free to customize it in any way you wish (the setup program will never overwrite your CMDUSER.CPM file). Many of the menu items in this file contain comments to help you follow along. At the top of this file is an initialization, or AUTOEXEC, section: this is where you can put any programs that you want to be run automatically when Command Post starts up, or any commands that you want to be executed upon startup.

CMDGROUP.CPM

This menu file contains one top-level menu item, titled "Groups". This **Groups** menu contains a single item: **Rebuild Menu from ProgMan Grps**. Selecting this item causes Command Post to read your Program Manager program group (.GRP) files and convert them to Command Post menu items, which it adds to the **Groups** menu. You can select **Rebuild Menu from ProgMan Grps** as often as you wish; each time you do, it will rebuild the entire **Groups** menu.

Command-line Parameters

Normally, when you start up Command Post, it loads the menu files listed as **Menu0** through **Menu9** in the [CmdPost] section of WWWPROD.INI. However, if you wish, you can run Command Post and specify the name of an alternate CPM file as a command-line parameter.

For example:

CMDPOST mystuff.cpm

This will cause Command Post (either the first instance or a secondary instance of Command Post) to start up using MYSTUFF.CPM as its menu file, and will *not* load any of the menus specified in WWWPROD.INI.

You can also specify the name of an application (or a data file which has been associated with an application) on the Command Post command line, such as:

CMDPOST calendar.exe

This will cause Command Post to start up (with its normal menus) and load the Windows Calendar program automatically. If you have made Command Post your Windows shell, then you can run Windows with a command line parameter, such as:

WIN calendar.exe

Which will also cause Command Post to load Calendar automatically on startup.

Replacing the Task Manager

Windows uses a special program called the **Task Manager** (TASKMAN.EXE). The Task Manager is invoked whenever you do any of the following:

1. Press Ctrl-Esc.
2. Select Switch To... from the system menu of any application.
3. Doubleclick on the screen background with the mouse.

The Task Manager supplied with Windows brings up a dialog box which lets you choose which window to go to.

Command Post gives you most of Task Manager's functionality. For this reason, you may wish to have Command Post act as your task manager, replacing TASKMAN.EXE. The procedure for doing this depends on the version of Windows you are using. Under Windows 3.0, it is as follows:

1. Rename Microsoft's TASKMAN.EXE to something different, such as TASKMANG.EXE.
2. Copy TASKMANR.EXE from the TASKMAN subdirectory (this subdirectory was created under the directory where you chose to install Command Post) to your Windows directory.
3. Rename TASKMANR.EXE to TASKMAN.EXE.

Under Windows 3.1 or higher, you can use the above method, or you can:

1. Copy TASKMAN.EXE from the TASKMAN subdirectory (this subdirectory was created under the directory where you chose to install Command Post) to your Windows directory.
2. Edit the "TASKMAN.EXE=" line in the [Boot] section of your SYSTEM.INI file to read as follows:
TASKMAN.EXE=TASKMANR.EXE
3. Restart Windows.

After that, invoking Task Manager (using any of the methods listed above) will simply cause Windows to switch to Command Post.

Windows' Task Manager has one feature that Command Post lacks: in some instances it lets you shut down otherwise recalcitrant programs with the End Task button. For this reason, you may want to make Windows' Task Manager available as a Command Post menu item:

```
&Utilities
  &Task Manager
    Run ("Taskmang.exe", "")
```

Limitations

Command Post supports a maximum of 100 hotkeys per window.

COMMAND POST DEFAULT MENUS

Control Menu

The **Control Menu** (also known as the **System Menu**) is the little box in the upper left hand corner of the Command Post window, to the left of the title bar. It is the only menu in Command Post which cannot be customized by the user. In addition to the standard menu items which are common to all Windows applications (**Restore, Move, Size, Minimize, Maximize, Close, and Switch To...**), **you will find the following:**

Enter License Info

(Appears only on shareware copies of Command Post.)

Use this selection to enter your license number and ID when you register your copy of Command Post. Registering brings you wonderful benefits:

- Gets rid of that pesky reminder window that comes up every time you start the program.
- Entitles you to one hour free telephone support for 90 days.
- Gets you the latest version of Command Post.
- Gets you your *own* printed copy of the documentation.
- Encourages the authors of this program to continue bringing you new and better products instead of breaking down and getting a real job.

Reload Menu

This is the selection you must choose after making changes to an active menu file in order to enable the changes.

About Command Post...

Displays copyright, version, and license information.

Terminate Windows

This menu item ends your Windows session.

(the second column)

The right-hand column of entries contains the names of all the open application windows. You can switch to one of these programs by doubleclicking on its title in the menu, or pressing the number which appears to the left of the title, or selecting the title and pressing **Enter**.

File Menu

Run...

Load...

These selections launch a program. **Run...** starts it as a "normal" window, i.e. not full-screen. **Load...** starts the program as an icon.

In either case a dialog box is displayed asking you for the name of the file you want to start. Whichever file is currently highlighted in the directory listing is the default.

Browse

This menu item runs the Command Post **Browser** program to view the selected file. See the documentation for Browser in Appendix C of the **WIL Reference Manual**, or use the on-line help built into Browser.

Edit...

Allows you to edit the selected file. The default editor is **Notepad**, but you can change this by selecting **Editor Options** from the **File** menu, under **Preferences**.

Copy...

This menu item lets you copy the selected files to another filename, directory, or drive. If you have more than one Command Post window open, the default destination is the second-most recently accessed Command Post window. If you have only one Command Post window open, a dialog box is displayed for you to specify the destination pathname; the default is the most recently-used copy or move destination, or the current directory if this is the first copy or move operation that has taken place.

Move/Rename...

This selection lets you rename the selected files to another name, or to move them to another directory or drive. If you are moving files, and have more than one Command Post window open, the default destination is the second-most recently accessed Command Post window. If you have only one Command Post window open, a dialog box is displayed for you to specify the destination pathname; the default is the most recently-used copy or move destination, or the current directory if this is the first copy or move operation that has taken place.

You can move *and* rename files in the same process.

Delete File...

This menu item lets you delete the selected files. A dialog box will appear to confirm you really want to delete them.

Print...

This copies the selected file to any standard (non-PostScript) printer. The first time you select this item, it prompts you for the device to print to, and stores that information for future use. If you ever wish to change this, select **Reset Assorted Options** from the **File** menu, under **Preferences**.

This command is meant to be used with text files only; it doesn't attempt to format the file or otherwise interpret it.

File Info

This menu item displays a message box showing the name, size, date, time, and attributes of the selected file(s), as well as showing how much space is taken up by the selected files.

Freespace on Local Drives

This menu displays a message box showing you how much space is available on your hard drives. The totals are given both in kilobytes and using a (rudimentary) bar graph.

Exit Windows

This menu item ends your Windows session.

Find file

On DOS Path

This prompts you for a file name, and looks for it in each directory on your DOS path. If successful, it displays a dialog box giving you the opportunity to change to the directory where the first occurrence of the file was found. Wildcards may not be used in the file name.

This operation takes no more than a few seconds.

On Current Drive

This prompts you for a file specification, which may contain wildcards, and looks for all occurrences of it on the current drive. If successful, it displays a box containing a list of all files that were found which matched the specification, and if you select one of the files in the list it will change to the directory where the file is located.

This can be a **very** long operation, especially on a large hard drive.

Hilite Files

Unhilite Files

These menu items are used to highlight or unhighlight all files in a directory which match a particular file specification (**By Name**) or which have a particular file attribute (**By Attribute**). They display a dialog box allowing you to enter either a wildcarded file specification (which defaults to all files with the same extension as the currently-selected file) or a file attribute (which defaults to files with the Archive attribute set).

File Attributes

Displays a dialog box allowing you to change the attributes of either the current file (**View/Modify current file**) or all selected files (**Change hilited files**).

Floppy Space

Displays a message box showing you how much space is taken up by the selected files and how much free space is available on your floppy drives (either **A:** or **B:**).

This is especially useful when you are preparing to copy files to a floppy disk and want to know if you have enough space to receive the files.

Networks

Network commands are part of the main file menu.

(These menu items are available only if you are running on a network.)

Connect Drive to Net

This menu item allows you to browse from a list of available network directories, and connects the selected directory to an available local drive letter.

Disconnect

This menu item displays a list of local drive letters which are connected to network directories, and disconnects the drive you select.

Net Dialog

This menu item displays the network's dialog box. The functions which are available from the dialog depend on the capabilities of your network.

Preferences Menu

These menu items allow you to change various Command Post options:

Show System/Hidden Files

Controls whether hidden and system files are listed in the Command Post file display. By default, such files are not displayed.

Save CmdPost Window Positions

Saves the positions of the currently-open Command Post window(s) for future use by the window-sizing logic which automatically positions all Command Post windows on open and close.

Auto Run= and Load= Options

Controls whether the programs listed on the "Run=" and "Load=" lines in the WIN.INI file are run automatically when Command Post starts up. This can be useful if you frequently switch between Command Post and another shell, such as **Program Manager**. By default, Command Post does not run programs listed in WIN.INI on startup.

CmdPost Window Titles

Allows you to specify the window titles for each of the four possible Command Post windows.

View Menu Startup Options

Controls whether the settings you make on the **View** menu are saved and used as the default settings when Command Post starts up in the future.

Editor Options

Allows you to specify an editor to be used by the **File/Edit** menu item. By default, **Notepad** is used.

File Refresh

Controls whether the Command Post file display is automatically updated after a program is run. By default, this option is on.

Default DoubleClick Program

Allows you to specify which executable program should be run when you try to launch a data file whose extension is not listed in the [Extensions] section of WIN.INI.

Reset Assorted Options

This menu item lets you reset the stored information about the types of floppy drives you have installed, the port to which your text printer is connected, and the directory where you keep your wallpaper files. The next time you select a menu item which makes use of this information, you will be prompted to re-enter it.

This is useful after making changes to your system configuration.

Extensions...

Windows gives you the ability to double-click on a data filename and automatically run a program with it based on the data file's extension. For instance, you can doubleclick on a .CRD file and the **Cardfile** program will start up and load the file you chose. The "**association**" between .CRD files and the Cardfile program is stored in the [Extensions] section of the WIN.INI file.

The **File/Extensions...** menu displays a dialog box allowing you to add or change the association between the selected file's extension and a program. If you change the association line, the new relationship will be recorded in the WIN.INI file. This is more convenient than having to bring up **Notepad** and edit WIN.INI in order to change the association.

The next time you doubleclick on that file (or any *other* file with the same extension), the associated program will be run with it.

Dir Menu

Create Directory...

This menu item lets you create a subdirectory under the current directory.

Rename Directory...

This menu item lets you rename the currently-selected subdirectory.

Kill Directory

This menu item deletes the selected subdirectory(s), if it is empty.

Change Directory...

This selection changes the current directory to one that you specify.

You can also change directories by doubleclicking on a subdirectory name, clicking on the pathname above the file listing, or pressing backspace to go back up to the parent directory.

Nuke ENTIRE file/dir structure

This menu item deletes all selected files and directories, as well as any files and subdirectories contained in the selected directory(s). It will display several warning messages before beginning the operation, but once it has started it does **not** prompt for confirmation of individual files.

Warning: This operation can permanently delete a great deal of data in a relatively short period of time. Make sure you know what you are doing.

Duplicate ENTIRE file/dir structure

This menu item copies the selected directory (including any files and subdirectories it contains) to a different location. It will prompt you for the name of the target (destination) directory before it begins. The source (original) directory and files are left untouched.

Move ENTIRE file/dir structure

This menu item moves the selected directory (including any files and subdirectories it contains) to a different location. It will prompt you for the name of the target (destination) directory before it begins. The source (original) directory and files are deleted.

Warning: Use this option with care.

Size of ENTIRE file/dir structure

This menu item calculates the total size of all selected files and directories, as well as any files and subdirectories contained in the selected directory(s).

Format Diskette...

This selection formats a diskette in one of your floppy drives by calling the MS-DOS **Format** program.

The first time you choose **Format Diskette...** after installing Command Post, you are asked what kind of floppy drives your system has. Command Post saves this information in the WIN.INI file for the next time you choose **Format Diskette...**. If your hardware changes, you can reset this information by selecting **Reset Assorted Options** from the **File** menu, under **Preferences**.

Directory Tree

When you select this menu item, a window is displayed which graphically shows the directory structure of the current drive. You can scroll through the structure, and then click on the directory you want to change to.

If you have two Command Post windows open, then clicking on a directory in the Tree with the **left** mouse button (or pressing the **Enter** key) causes the **left-most** Command Post window to change to the

selected directory, and clicking on a directory in the Tree with the **right** mouse button (or pressing **Shift-Enter**) causes the **right-most** Command Post window to change to the selected directory.

If you have only one Command Post window open, then clicking on a directory in the Tree with the **left** mouse button (or pressing the **Enter** key) causes the Command Post window to change to the selected directory, and clicking on a directory in the Tree with the **right** mouse button (or pressing **Shift-Enter**) causes a second Command Post window to open with the selected directory displayed.

View Menu

Short

Long

These selections control how much information is displayed for the files in the Command Post window.

View/Short displays only the filenames, while **View/Long** also displays the size, the date/time modified, and the file attributes.

All

Partial...

Programs

These selections determine which filenames are shown. **All** displays all files in the current directory.

Partial... displays a dialog box where you specify one or more wildcarded filenames to show.

Programs shows only those files with an extension of .BAT, .COM, .EXE, or .PIF.

By Name

By Date

By Size

By Kind

Unsorted

These selections specify how to sort the files in the display. The default is to sort by **Name**. Sorting by **Date** shows the most-recently modified files before the older ones. Sorting by **Size** shows the largest files first. Sorting by **Kind** displays the files by their extensions. **Unsorted** produces a list in the same order as you would get from a **dir** listing in MSDOS.

If you have the file display set to **View/Long**, then whichever column is being sorted on is displayed in a bold font.

Window Arranging

1 Stack

2 Arrange

3 Arrange in Rows

4 Arrange in Columns

5 (w/o CP) Stack

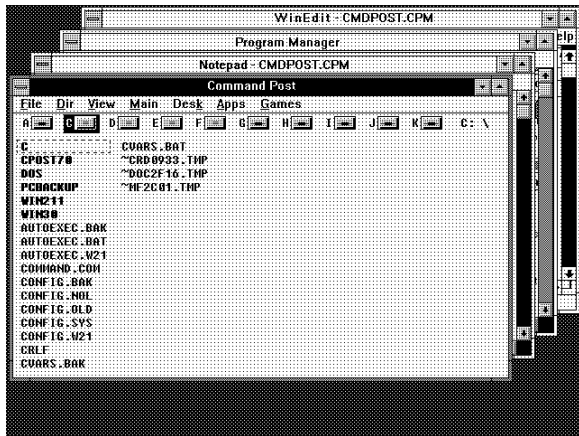
6 (w/o CP) Arrange

7 (w/o CP) Arrange in Rows

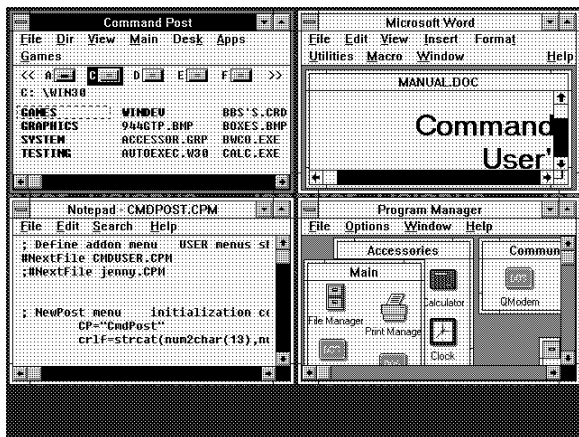
8 (w/o CP) Arrange in Cols

The right column of the View menu contains the window arranging functions. There are four ways Command Post can arrange application windows on the screen for you:

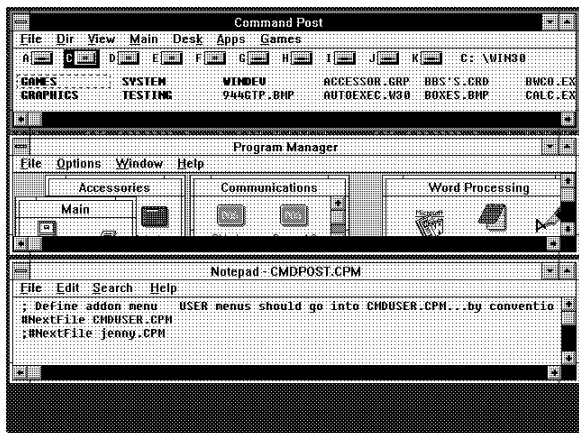
1 Stacked



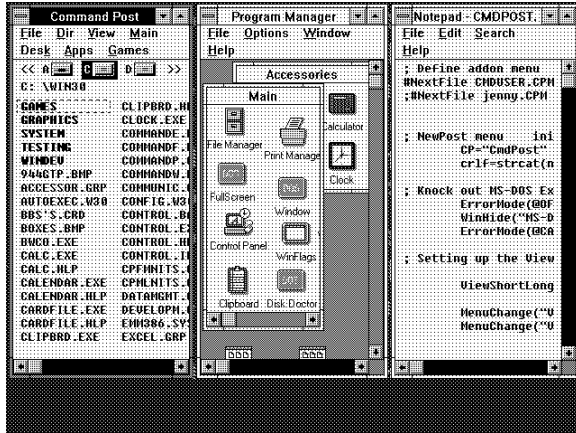
2 Arranged (Tiled)



3 Arranged in Rows



4 Arranged in Columns



Selections **5** through **8** are the same as **1** through **4**, except that the Command Post window is minimized.

When you specify **Arrange in Rows** and you have more than four open windows, or if you specify **Arrange in Columns** and there are more than three open windows, Command Post will revert to **Arrange**.

Change Wallpaper

This menu item replaces the current background Wallpaper with a new Wallpaper (BMP) file of your choice.

Blank Screen Now!

This menu item immediately activates the Windows 3.1 screen saver that you have configured through the **Desktop** section of **Control Panel**.

Main Menu

Command Post (More Windows)

This menu item opens a second, third, or fourth Command Post window. This way you can view more than one directory at a time. Also, when you choose **File/Copy...** or **FileRename/Move...**, the destination pathname defaults to the "other" Command Post's directory.

Command Post (Show All Windows)

This menu item repositions all open Command Post windows to their default coordinates, which you can change by selecting **Save CmdPost Window Positions** from the **File** menu, under **Preferences**.

Control Panel loads the Windows Control Panel.

Clipboard loads the Windows Clipboard viewer.

DOS Prompt

This selection runs COMMAND.COM to enable you to work in DOS directly. You can press **ALT+ESC** to switch back and forth between Windows and DOS, and when you are through using the command-line interface you type "**exit**" from the DOS prompt to return to Windows.

Run "Managers"

Program Manager

File Manager

Print Manager

Edit Configurations

Edit CmdPost menus

This menu item displays a listbox containing all the .CPM files in the Command Post directory, and launches **Notepad** with the file you select.

After you edit a Command Post menu file, you can either select **Reload Menu** from Command Post's control menu to have the changes take effect immediately, or you can wait for the menus to be reloaded automatically the next time you make a selection from the menu. However, in the latter case you will get a message telling you to re-select the menu item.

Load CmdPost Help File

Load WIL Help File

Edit INI files

This menu item lists all the .INI files in the Windows directory, and allows you to select one to be modified. It then lists all the sections contained in the desired file, and after you select a section it lists all the lines contained in that section, and, finally, after you select a line, it allows you to edit that individual line.

Note: This menu item cannot be used to edit the "Device=" lines which appear in the **[386Enh]** section of the SYSTEM.INI file.

Run System Configuration Editor

This selection runs the Windows **System Editor**. The **System Editor** allows you to edit the AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, and SYSTEM.INI files.

Run Windows Setup

This menu item runs the Windows **Setup** program. This lets you review your current Windows setup configuration and change it if necessary.

Restart Windows

This menu item gives you the following three options:

NO! Wait

This option doesn't do anything (other than display a brief message).

Really! Doit!

This option restarts Windows, just as if you were to exit Windows and then manually start it up again from DOS.

Reboot! this machine

This option performs a complete warm boot of the computer, just like pressing **CtrlAlt-Del** twice. It only works under Windows 3.1 or higher; under Windows 3.0, it simply restarts Windows.

Warning: Use this option with care.

PIF Edit...

This menu item lets you edit a Windows PIF file using the **PIF Editor**. If the currently-selected file is a PIF file, then the PIF Editor will be launched with that file. Otherwise, the menu item will display a listbox containing all the .PIF files in the current directory, and will launch the **PIF Editor** with the file you select. However, if there are no PIF files in the current directory, then the menu item will attempt to locate your PIF directory (by searching for _DEFAULT.PIF), and will list the PIF files in that directory.

Note: You can force this menu item to list files in your PIF directory (instead of the current directory) by holding down the **Shift** key while selecting PIF Edit... from the menu.

System Information

This selection shows a message box containing numerous pieces of interesting information about your Windows system configuration.

Interactive WIL

The items on this menu item allow you to interactively enter Command Post commands, and to recall previously-entered commands without having to re-type them. This is sometimes known as a "command stack":

Execute WIL Function

Allows you to interactively enter a Command Post function or command. The return value of the function will be displayed in a message box.

Execute a Previous Entry

Allows you to select and re-use a previous entry from the command stack.

Flush Previous Entries

Allows you to clear all entries from the command stack.

Command Post functions

Introduction

This section includes only those additional Command Post functions which do not appear in the **WIL Reference Manual**. The **WIL Reference Manual** is your primary reference to the functions available in Command Post.

Note: The functions listed under the **See Also** headings may be documented either in this User's Guide or in the **WIL Reference Manual**.

Function List

DirExist ([d:]path)

Determines if a directory exists.

DiskHide (drive-list)

Hides disk drives from display.

DiskReset ()

Re-examines available disk drives.

DiskUpdate ()

Updates drive icon display.

FileHilite(file-masks, mode)

Highlights or unhighlights files in file display.

IntControl(request#, p1, p2, p3, p4)

Internal control functions.

MsgTextGet(window-name)

Returns the contents of a Windows message box.

OtherDir ()

Finds the directory where another copy of Command Post is running, if any.

OtherUpdate ()

Updates another Command Post directory display.

Refresh

SetDisplay(detail, sort-by, masks)

Controls the display of files in the Command Post window.

DirExist

Determines if a directory exists.

Syntax:

DirExist ([d:]path)

Parameters:

(s) [d:]path directory name, with optional drive.

Returns:

(i) **@TRUE** if the directory exists;

@FALSE if it doesn't exist.

You can use this function to determine whether a specified drive is valid by checking for the existence of the root directory on that drive.

Examples:

```
wmdir = "c:\wp"
If DirExist(wmdir) == @FALSE Then DirMake(wmdir)
DirChange(wmdir)

:top
drive = AskLine("Run Excel", "Enter a drive letter", "")
If drive == "" Then Exit
drive = StrSub(drive, 1, 1)
If DirExist("%drive%:\") == @FALSE Then Goto top
NetAddCon("\\userapps\excel", "", drive)
```

See Also:

[DirChange](#), [DirMake](#), [DirRemove](#), [DirRename](#), [FileExist](#)

DiskHide

Hides disk drives from display.

Syntax:

DiskHide (drive-list)

Parameters:

(s) drive-list string of drives to hide (non-delimited).

Returns:

(i) always 1.

This function causes the drive letters specified in **drive-list** to be removed from the disk drive icon display.

Example:

```
DiskHide("STUVW")
```

See Also:

[DiskReset](#), [DiskUpdate](#)

DiskReset

Re-examines available disk drives.

Syntax:

DiskReset ()

Parameters:

(none)

Returns:

(i) always 0.

Examines disk drives on system, and adds any new drives found to the display of drive icons. If an existing drive was hidden with the DiskHide function, it will no longer be hidden (unlike the DiskUpdate function).

Example:

DiskReset ()

See Also:

DiskHide, DiskUpdate

DiskUpdate

Updates drive icon display.

Syntax:

DiskUpdate ()

Parameters:

(none)

Returns:

(i) always 0.

Examines disk drives on system, and adds any new drives found to the display of drive icons. If an existing drive was hidden with the **DiskHide** function, it will remain hidden (unlike the **DiskReset** function).

Example:

```
DiskUpdate ( )
```

See Also:

[DiskHide](#), [DiskReset](#)

FileHilite

Highlights or unhighlights files in file display.

Syntax:

FileHilite (file-masks, mode)

Parameters:

(s) file-masks one or more file specifications, which may be wildcarded.

(i) mode @TRUE Highlight matching files.

@FALSE Unhighlight matching files.

Returns:

(i) total number of files highlighted or unhighlighted.

This function causes one or more groups of files in the file display window to be highlighted (selected) or unhighlighted (de-selected). This is useful to select files for an operation such as **FileCopy**, or just to spotlight certain files in a directory.

Multiple file specifications must be space-delimited.

Examples:

```
FileHilite("*.ZIP *.LZH *.ARC", @TRUE)
```

```
FileHilite("OLD*.ZIP", @FALSE)
```

See Also:

[CurrentFile](#), [FileExtension](#)

IntControl

Internal control functions.

Syntax:

IntControl (request#, p1, p2, p3, p4)

Parameters:

- (i) request# specifies which sub-function is to be performed (see below).
- (s) p1 - p4 parameters which may be required by the function (see below).

Returns:

- (s) varies (see below).

Short for Internal Control, a special function that permits numerous internal operations. The first parameter of IntControl defines exactly what the function does, the other parameters are possible arguments to the function.

Refer to the **WIL Reference Manual** for information on other sub-functions.

IntControl (2, 0, 0, 0, 0)

Returns the number of Command Post windows currently open.

IntControl (3, 0, 0, 0, 0)

Writes the positions of each open Command Post window to the WWWPROD.INI file, using the WinPositionXY format.

IntControl (6, 0, 0, 0, 0)

Positions all open Command Post windows, based on the information in the WWW-PROD.INI file.

IntControl (9, p1, 0, 0, 0)

Controls Command Post window resizing.

| <u>P1</u> | Meaning |
|-----------|--|
| 0 | Resize automagically on open and close (default) |
| 1 | disable resize on window close |
| 2 | disable resize on window open |
| 3 | disable resize on open and close |

IntControl (11, p1, 0, 0, 0)

Used to tell Command Post that it is (or is not) a shell, contrary to what it really is. That is, if it is really a shell, you can disable the shell-like characteristics, or if it is not a shell, enable its shell characteristics.

However, if you are using Windows 3.1 or higher, and Command Post is your Windows shell, it is generally recommended that you do **not** use this function to tell it otherwise.

| <u>P1</u> | Meaning |
|-----------|-------------------|
| 0 | Play standard app |
| 1 | Play shell |

MsgTextGet

Returns the contents of a Windows message box.

Syntax:

MsgTextGet (window-name)

Parameters:

(s) window-name full title of the message box window.

Returns:

(s) contents of the message box.

This function returns the text contents of a standard Windows message box. "Window-name" must be the full title of the message box window, and is case-sensitive.

Note: This function will not work with the types of message boxes created by most WIL functions, since they are not standard Windows message boxes.

Example:

```
msg = MsgTextGet("Microsoft Word")  
If msg == "Search text not found" Then SendKey("~")
```

OtherDir

Finds the directory where another copy of Command Post is running, if any.

Syntax:

OtherDir ()

Parameters:

(none)

Returns:

(s) the directory of the second-most recently used Command Post window. The current window is considered the most recently used window.

Use this command to determine the directory of the second-most-recently-used Command Post window.

Useful in setting up copy and move operations between two instances of Command Post.

Example:

```
a = DirGet()
```

```
b = OtherDir()
```

```
Message("Directory of this CmdPost window is", a)
```

```
Message("Directory of the other CmdPost window is", b)
```

```
Drop(a, b)
```

See Also:

[DirGet](#), [DirHome](#), [OtherUpdate](#)

OtherUpdate

Updates other Command Post directory displays.

Syntax:

OtherUpdate ()

Parameters:

(none)

Returns:

- (i) **@TRUE** if another copy of Command Post was found to update;
@FALSE if this is the only copy running.

This command updates the file displays of any other open Command Post windows. This is useful if your menu item changes a directory; i.e. if a file or directory is created, moved, renamed, or deleted. **OtherUpdate** helps ensure the other Command Post display(s) immediately reflect the change the user caused from *this* copy.

Example:

```
FileCopy("myfile.txt", OtherDir(), @FALSE)  
OtherUpdate()
```

See Also:

[OtherDir](#), [Refresh](#), [SetDisplay](#)

Refresh

Updates file window display.

Syntax:

Refresh (request#)

Parameters:

(i) request# see below.

Returns:

(i) always 1.

This function updates the files being displayed in the Command Post window, as well as the drive icons that are displayed. It is useful after running an application (such as a DOS program) which creates, deletes, or renames a file in the directory being viewed, or connects to (or disconnects from) a network server, and you wish the changes to be reflected in the Command Post window.

Specifying a request# of 0 causes only the active Command Post window to be updated, and specifying a request# of 1 causes all Command Post windows to be updated.

Refresh(0) is the same as **SetDisplay("", "", "")**, and **Refresh(1)** is the same as **OtherUpdate()**.

Note: This command does not take effect until the WIL program has completed, regardless of where the command may appear in the program.

Example:

```
Run("pkunzip.exe", CurrentFile())  
Refresh(0)
```

See Also:

[OtherUpdate](#), [SetDisplay](#)

SetDisplay

Controls the display of files in the Command Post window.

Syntax:

SetDisplay (detail, sort-by, masks)

Parameters:

- (s) detail level of detail. Use "**SHORT**", "**LONG**", or "**NONE**".
- (s) sort-by how to sort the filenames. Use "**NAME**", "**KIND**", "**SIZE**", "**DATE**" or "**UNSORTED**".
- (s) masks list of masks for file display.

Returns:

- (i) **@TRUE** if valid options were specified;
@FALSE if invalid.

Use this function to change and/or update the file display.

If "detail" is set to "SHORT", only file names and extensions will be displayed. If it is set to "LONG", file sizes, times, and attributes will be displayed as well. If it is set to "NONE", there will not be any file display at all; this is useful for security reasons, to prevent users from executing unwanted programs.

Any of the fields may be null. If a field is null the previous setting is used. This command will alter the file display Parameters, and then re-read all the files and update the display.

A special form of this function, **SetDisplay ("", "", "")**, will update the file display without changing any of the previously set parameters. This is necessary after changing directories if you want the new directory to be shown, and is useful after running programs which create or delete files.

By default, hidden and system files are not displayed. This can be changed by setting "SysHide=Yes" in the [CmdPost] section of WWWPROD.INI.

Note: **SetDisplay** does not take effect until the current WIL program has completed, regardless of where the command may appear in the program.

Example:

```
Windows &SDK
&Show SDK Development Files
SetDisplay("", "", "*.ICO *.CUR *.BMP *.DLG *.H")
```

See Also:

[OtherUpdate](#), [Refresh](#)

UTILITIES

Dialog Editor

The WIL Dialog Editor (WWWDLGED.EXE) provides a convenient method of creating dialog box templates for use with the **Dialog** function.

It displays a graphical representation of a dialog box, and allows you to create, modify, and move individual controls which appear in the dialog box.

After you have defined your dialog box, the Dialog Editor will generate the appropriate WIL code, which you can save to a file or copy to the Clipboard for pasting into your WIL program.

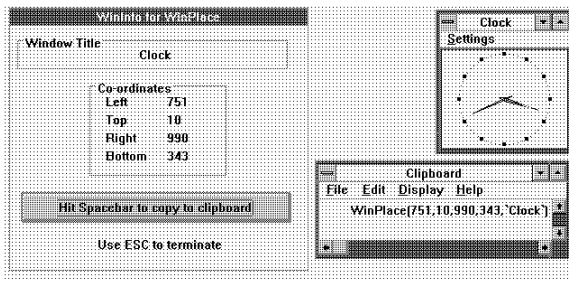
The WIL Dialog Editor comes with an online help file (WWWDLGEDT.HLP). Simply select the **Help** function in the Dialog Editor for detailed instructions on using the program.

If the file WWWDLGED.EXE is located in the FIL-CMDR.HLP directory, you can try it by clicking the button here. You can also load it from the menu button bar at the top of this screen.



WinInfo

The **WinInfo** utility (WININFO.EXE) lets you take an open window that is sized and positioned the way you like it, and automatically create the proper **WinPlace** statement for you. It puts the text into the Clipboard, from which you can paste it into your WIL program:



You'll need a mouse to use **WinInfo**. While **WinInfo** is the active window, place the mouse over the window you wish to create the **WinPlace** statement for, and press the spacebar. The new statement will be placed into the Clipboard. Then press the **Esc** key to close **WinInfo**.

If the file WININFO.EXE is located in the FIL-CMDR.HLP directory, you can try it by clicking the button here. You can also load it from the menu button bar at the top of this screen.



