

WinCmd v3.0
A Task Dispatcher for Microsoft Windows

WinCmd Help Index

WinCmd General Information

Sample Config File

The Control Button

Procedures

Building a Config File

Starting a Program

Config Summary

Re-Building WinCmd in Windows

Introduction

WinCmd is a menu builder application for Microsoft Windows that allows any user to build pop-up Menus, prompt for input and launch applications to either Windows or DOS simply by clicking a Button. The WinCmd [configuration file](#) which builds the Menus and Buttons is a simple ASCII text file and can be created using any text editor under DOS or Windows. Using WinCmd, you can easily

- * build one or many pop-up Menus linked to custom push-buttons.
- * add Sub-Menus to any pop-up Menu.
- * build as many items as you need in each Menu (max is 99 total)
- * specify ALT-key combinations or Hot-keys for any Menu or Button.
- * add Separator lines in your Menus.
- * specify any DOS or Windows file (EXE, COM, BAT, PIF etc) to be executed.
- * prompt for User Input from any Menu item.
- * build Dialog boxes and List boxes.
- * supply default Response to any user input if desired.
- * change and rebuild your Menus on-line using Notepad (or any editor).

WinCmd goes beyond batch files or the Program Manager "Run" box by allowing user-input to modify an application launch and providing the ability to supply defaults. With WinCmd you decide what your menus will look like and how tasks are grouped.

Unlike Program Manager groups, where the command line is fixed, WinCmd allows you to change the parameters you pass to any application at will, and shields the user from the need to know path-specific information in order to start a task.

Starting a Program

You can launch a DOS or Windows program from WinCmd by simply pushing a Button in the main window.

Using the mouse

Click the Mouse left button on the chosen WinCmd button. A pop-up menu will appear to one side of the main WinCmd window. Click on the menu item you want to execute.

Using the Keyboard:

WinCmd buttons carry a name, of which one letter will appear underlined. To select a button, press the Alt key in combination with the underlined character. A dotted selection box will appear around the button's title. To activate the button, press the Space Bar. A pop-up menu will appear to one side. Use the arrow keys to move in the pop-up menu. Press Enter to select an item to execute.

Building the WinCmd.CFG file:

Use any text editor to build or modify the config file to your needs.

There are 5 types of cfg file entries.

the Title Bar which sets the WinCmd title.

the Button Name which sets the name of the push-button in the main window.

a Menu Separator to insert a separator line in a Pop-Up menu.

a Command Line to define what is to be executed.

a Sub-Menu item to create a pop-up submenu inside a menu.

see Sample Config File

Title Bar

The first line of the Config file can be a Title text to appear in the Title Bar of the WinCmd window. This Title entry is optional. The first character of this line is a pound sign (#), and the remaining text becomes the title.(the '#' is not shown).

Only one Title entry is allowed in a config file, and if used , it must be the first line in the file.

see [Sample Config File](#)

Menu Button Name

Any line that begins with an exclamation (!) symbol is the title of WinCmd Button.

A Button with this title text will be displayed in the WinCmd window, and will pop-up a menu of following items when the button is pushed.

The ampersand (&) char preceeding another char on the line will cause the char to be shown underlined in the button title, and be treated as an ALT-key combination by windows.

ie. SA&VE will appear as SAVE, the "V" underlined. ALT-v will select it.

see Sample Config File

Separator Line

Any line containing a tilde (~) char only will create a Separator line in the pop-up Menu displayed when a button is pressed.

Separators are useful for visual grouping of items in a pop-up menu.

They serve no logical function in Windows.

EG: Item 1, cmd 1,
 ~ <----- Separator line symbol
 Item2, cmd2
 Item3 , cmd 3 etc

Command Line

A Command line contains up to four fields, each separated by a comma.

The general format of a Command Line is....

Menu Item , DOS Command , User Prompt , Default Response

Spaces between the fields are ignored by WinCmd.

The Menu Item and Command fields are required.

The User Prompt and Default Response fields are optional.

A special type of Command line, with a single Caret (^) in the Command field, is used to specify a Sub-Menu.

see Sample Config File

Menu Item

This is the name of the Menu Item and will be displayed in the pop-up Menu when it is selected from a button. Spaces are allowed in the name

The **&** character acts to underline the following char and assigns a hot-key to it.

eg S&econd Choice is shown as Second Choice, the 'e' underlined, and is selected by pressing the 'e' key.

eg: **Data Base** , c:\util\db.exe, Enter file to use ?, myfile.dat

see Sample Config File

Command

This is the Command to be executed when an item is selected by the user from the pop-up menu. It can be the name of any DOS or Windows executable file and can include path information and parameters .

The command should include a file extension. (exe ,com,bat,pif)

It cannot be a DOS internal command, such as DIR, but can be a DOS external command (eg XCOPY.EXE)

eg: Data Base , **c:\util\db.exe** , \data\myfile.dat

would launch DB.EXE to operate on MYFILE.DAT in directory \DATA

A single Caret (^) is a special case for defining Sub-menus...

see Sample Config File

User Prompt (optional).

If present, will cause WinCmd to prompt for user input.

The Prompt box will be built with the text in this field and displayed when the user selects this item from the pop-up menu.

The user's response is passed to Windows as a parameter.

eg: Data Base , c:\util\db.exe, **Enter file to use ?**

would display a prompt box with "Enter file to use ?" as the question to the user.

If the Prompt field ends in ?? it is a special case for building a List Box .

see Sample Config File

Default Response (optional).

If used, is the default response to the User Prompt field

eg: Data Base , c:\util\db.exe, Enter file to use ?, **myfile.dat**

would display a prompt box with the response field pre-loaded with the text "MYFILE.DAT" .

The user may accept this or change any or all of it by supplying a new response.

If the Response contains a DOS wildcard character, (*,?) it is a special case. for building a List Box .

see Sample Config File

Building a WinCmd List Box

If the User Prompt field ends in 2 question marks (??)

OR

If the Default Response field contains a wildcard filespec (*.*, *.txt, ?ltr.doc etc)

a List Box will be built to allow the user to pick a filename response.

The List Box will be shown instead of the normal WinCmd Prompt box.

Example 1 Data Base , c:\util\db.exe, **Enter file to use ??**, myfile.dat

Example 2 View a File, notepad.exe, File to view ?, **\letters*.doc**

WinCmd will build a List Box displaying all the .DOC files in the directory \letters.

The user can pick a file or view another filetype or even a different directory and/or drive. Double clicking on the file or clicking the "Select" button will pass the selection back to WinCmd as a parameter to the command line.

Clicking the "Cancel" button closes the List Box and sends a Null response to WinCmd.

see Sample Config File

Sub-Menu

A Command line beginning with the Caret '^' character in the Command field denotes the start of a Sub-Menu.

The line should have text only in the Title and Command fields. Other fields are meaningless.

All following lines which begin with '^' are attached as sub-menu items, with a format identical to a normal Command line .

example:

!Phone Data	<-----Appears in Header Bar (starts with !)
All Employees, \wscr\notepad.exe c:\text\empall.lst	<--normal CMD line
Employee Phone# , ^	<-----Start of Sub-Menu (command field = ^)
^Names A to F ,notepad.exe emp1.lst	<--1st Sub-Menu item
^Names G to Q ,notepad.exe emp2.lst	<--2nd Sub-Menu item
^Names R to Z ,notepad.exe emp3.lst	<--3rd Sub-Menu item
^Any Name, notepad.exe, Which Group?,emp*.lst	<--last Sub-Menu item
Customer ,\wscr\notepad.exe c:\text\cust.lst	<--normal CMD line

Summary

In summary:

- # marks Title text, to be shown in the Windows title bar.
 - ! marks the start of a Menu, and is shown in a WinCmd button.
Following lines, up to the next line beginning with !, are Command lines belonging to this Menu.
 - ^ marks the start of a Sub-Menu. All following lines beginning with ^ are treated as attached sub-menu Command lines.
The first line that does not begin with '^' ends the sub-menu.
- Lines that begin with none of the above are normal Command lines.

The **.CFG** file is pretty much free-form.

Leading whitespace is ignored in all fields.

All fields are comma-separated.

If a field is optional no delimiters or placeholders are needed if following fields are also unused.

The maximum total number of entries in the .cfg file is 99.

A sample WinCmd.cfg file is supplied

Rebuilding WinCmd dynamically

To add, delete or alter any aspect of WinCmd it is only necessary to edit the **.CFG** file. It is possible to add or remove Buttons, menu items, sub-menus, the Title etc etc etc.

Using the current .CFG file:

Select the Control Button

Select "View/Edit Config". This will invoke Notepad to edit the current .CFG file.

Make the required changes, then save the .CFG file and exit Notepad.

Select the Control Button.

Click on "Rebuild". A prompt box will ask for a .CFG filename. The current .CFG will be shown as the default. Click OK

WinCmd will rebuild itself with the new configuration.

Using an alternate .CFG file:

Select the Control Button.

Click on "Rebuild". A prompt box will ask for a .CFG filename. The current .CFG will be shown as the default. Enter the new filename, then click OK .

WinCmd will rebuild itself with the new configuration file

The Control Button

is always built by WinCmd.

View/Edit Config: This launches Notepad to allow you to display and/or make changes to the current cfg file. (Notepad is an application supplied with Microsoft Windows). The current .CFG file is either WinCmd.cfg or the file specified when WinCmd was started.

Rebuild: WinCmd will prompt for the name of a cfg file to use, and then re-configure itself using the information supplied by the specified file. Use this after making changes with "View/Edit Config."

Quit: As the name implies this will terminate WinCmd.

Exit Windows will terminate Windows after prompting for confirmation.

Sample WinCmd Config File

```
#MyTitle <<------(the Title Bar text.)
!&Utils <<------( Button name)
 &View Directory , msdos.exe <<------(normal Menu Item )
 &Edit a File, notepad.exe,File to edit ?,wncmd.doc
~ <<------( Separator symbol)
View a File,^
 ^&Any file , c:\winapp\notepad.exe ,File to view ??, *.*
 ^&Txt file , notepad.exe ,File to view ?? ,c:\text\*.txt
~
 ^&Doc file , Notepad.exe ,File to view ? ,c:\text\*.doc
 ^&View ? file , Notepad.exe ,File to view ? ,c:\text\?able.doc
~
View &named file , Notepad.exe ,File to view ?
!Fax
 &Write FAX , notepad.exe
 &Send FAX via PC , c:\com\sndfax.bat , FAX File to send ?
 &Rec FAX via PC , c:\com\rcvfax.exe
!COMM
File &Xfer , c:\com\fx.bat ,File to send ? , emp.lst
 &Compuserve , c:\cserve\cim.exe
!&Phone# <<------( Button name )
 &All Emploeyess, Notepad.exe c:\text\empall.lst
Employee Phone# , ^ <<------(begin sub-Menu)
 ^Names &A to F ,Notepad.exe c:\text\emp1.lst <<------(1st sub-menu item)
 ^Names &G to Q ,Notepad.exe c:\text\emp1.lst
 ^Names &R to Z ,Notepad.exe c:\text\emp1.lst
~ <<------(sub-Menu separator)
 ^Any &Name, Notepad.exe, Which Group?,\text\emp*.lst
 &Customer ,Notepad.exe c:\text\cust.lst <------(end sub-Menu)
~
 &Choose entry ,Notepad.exe c:\text\cust.lst , Enter Name ?
!&List
View &Prog Info,Notepad.exe wncmd.doc
```

