PcMenu

Version 2.1

Reference Manual

Legal Stuff

Grant of Rights

The Author of PcMenu V2.1 hereby grants you the right to use this software at no charge, with exception to the restrictions below.

Restrictions

You may not **Sell** this software without the written permission from the author.

Limitations of Liability

This software is distributed AS IS, and the author cannot be held responsible, and will not be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out of your use of this software, even if the author has been advised of the possibility of such damage.

Redistribution

You may redistribute this software providing that, No Profit is attained, and, that this notice and all of its accompanying material, including all files and programs, are distributed together, whether it be by electronic media, or Bulletin Board Service.

THE USE OF THIS SOFTWARE INDICATES YOUR ACCEPTANCE OF THE TERMS OUTLINED ABOVE !

Comments

For those users who have Internet access, you can send your comments, sugestions and bug reports to PcMenu@arcadia.mc.phillip.edu.au.

Credits

This software was written by Brian Habel, a computer programmer who is employed at the Bundoora Campus of the Royal Melbourne Institute of Technology.

NOTICE

NOVELL and NETWARE are registered trademarks of Novell Inc. Provo Utah U.S.A

Introduction

What is PcMenu

PcMenu is a software package which allows System Administrators, Managers of Local Area Networks, or even the lone home user of a standalone system to organise the software on there system so that it can easily accessed.

This is achieved by creating customised screen menu's from a series of menu scripts, that are small ascii text files which define the contents of each menu. A menu can lead to other menus, referred to as sub-menus, allowing the user or administrator to organise software in a variety of ways.

PcMenu is fully compatible with NOVELL's NETWARE MENU utility 'menu script' files. Users of PcMenu in a NOVELL Network environment can easily install PcMenu as the scripts from NOVELL's menu utility work in the same way. Although PcMenu may be NOVELL compatible, it has even more to offer, as such storing sub-menus in other files from where they are called. Also, an optional screen saver has been added to reduce the chance of 'burn on'. PcMenu even has mouse support! With the Paldef utility you can even choose what colour you want for your menus.

PcMenu runs options in DOS via Batch Files with no memory overhead ! Yes, you can even load Terminate and Stay Resident (TSR) type programs with PcMenu. Why, I hear you ask? Simply, because PcMenu does NOT shell as it runs its options, it runs them under the Primary Command Processor.

You may like to think of PcMenu as a batch file database and each menu option is the link to its associated batch file. Now there is no need to have plethora of batch files roaming your system as all files can be integrated into one!

How To Use PcMenu

Planning your menus

Some thought must be considered in what a menu should contain. This all depends on who is going to be using the system and what environment the user is in. For example, a user in an educational environment may be using PC's connected via a local area network. This may mean that the user may need to access various amounts of information from various software packages. One plan might be to group all common software packages into groups, say all 'Word Processors' in one group, and all 'Spread Sheets' in another. These groups could then become each in them selves a sub-menu which can then be accessed from a main menu. Another plan might be to sort the software packages on a most used basis. Placing options for these in the first main menu, and other less frequently used software packages in sub-menus.

Defining your Menus

As mentioned before, PcMenu uses the same technique as the NOVELL NETWARE Menu utility. Those users who are familiar with the NOVELL NETWARE Menu utility should have little difficulty in designing and implementing menu's scripts for PcMenu.

Essentially, each menu has a Title and at least one option up to an upper limit of fifty. A complete menu definition has the following form.

%menu_title[,[row_placement][,[column_placement][,palette]]]
Menu Option
|
Dos Batch file commands
|
OR
%sub-menu_title[,alternative_filename]

Looking at this in more detail, each menu definition starts with the % character which <u>must appear in the first column of the file</u>, followed directly by the title of this menu. The title here will be the title that is displayed in the title area of your menu. Following the title there are several optional values that may be added for this menu. The first two of these are associated with the location on the screen you want the menu to be displayed and the third option is the palette number which defines the colour of the menu. These options will be discussed in more detail later.

On the following lines appear the options that appear in the menu. Each option that you wish to add, must again start in the first column of the file and as before, what you type here is displayed as your option in the menu. The next lines are what action is to be taken for this option. They usually consist of standard DOS batch file commands. Each Batch file command must be offset by at least a space. That is it must start in at least the second column of the menu definition file. You may find by using the tab character to space DOS batch file commands, you can keep your menu options more distinct from each other. PcMenu will continue reading batch file commands for this option until it finds the next option title, which should be located in the first column.

Alternatively, instead of a batch file command, you can place a % character and immediately following, the name of another menu definition. This creates a link to what is known as a sub-menu. You can have sub-menus located in a different file if you wish, in which case you add after the sub-menu name the complete DOS PATH of the filename in which this sub-menu is contained separated by a comma.

Example:

```
%Brian's Computer
Word Processors
	%Word Processors Menu
Dos Command
	echo off
	@"Enter your DOS command"
%Word Processors Menu,,,2
Word Perfect 5.1
	echo off
	c:\wp51
	wp
	cd c:\
PC  Write
	echo off
	cd c:\pcwrite
	ed
	cd c:\
```

This menu definition file contains a main menu (entitled 'Brian's Computer') and one sub-menu (entitled 'Word Processors Menu'). The main menu uses default values for menu location and palette colour. The sub-menu uses default the menu location but uses the user defined palette colour 2. The main menu would display something like the following.

```
Brian's Menu

Word Processors
PCWrite
```

Fig 1. Example Menu

Notice that the menu title is centred on the screen. If you do not want the title centred , it must be enclosed in double quotation marks,. The menu title will then be left justified.

Also note that menu items are not sorted alphabetically. Menu items appear in the menu in the same order they appear in the menu definition file.

IMPORTANT
Sub-menus located in alternate files will only load if that file is accessible. If the alternate file cannot be found or read THE ENTIRE MENU ENTRY FOR THIS OPTION WILL NOT BE DISPLAYED. That is, if PcMenu cannot load a sub-menu from an alternate file then it will ignore the entire option from the current menu definition when displayed. (If you cant read the option definition, why display an option to the user). On the other hand if the sub-menu definition is made and no alternate file is allocated, the sub-menu definition must be located in the current file otherwise an error will result.

Network administrators, might see some exploitation in this feature!

Locating Your Menus On The Screen

By default, if there are no optional values supplied after the menu title name for a menu definition, the menu will be displayed in the middle of the screen. To place the menu in a different location you supply two values which adjust the horizontal and vertical position of the menu. Lines are measured from the top of the screen to the centre of the menu in the horizontal plane, and from the sides of the screen to the centre of the menu in the vertical plane. So to calculate the final position of your menu apply the following formula.

$$A + B/2 = C$$

where

A = the number of lines above the menu

B/2 = half the number of lines in the menu counting the borders and title

C = row_placement value

and for the column_placement value re-use the same formula above except this time A, B, and C represent:-

A = the number of columns to the left of the menu

B = half the column width of the menu

C = column_placement value

If you make the placement values too large or too small the menu will be positioned at the appropriate edge of the screen. You need not worry about moving the menu off the screen. If the placement values are ZERO or omitted they revert back to default values, the centre of the screen for that plain (horizontal / vertical) .

Using Your Mouse With PcMenu

PcMenu supports any Microsoft Compatible mouse conected to your system!

While the mouse pointer (indicated by the red block) is over a menu option, pressing the left button once will move the highlight bar to that option. Pressing the mouse button twice in rapid succession will select that option. (As if the Enter key had been pressed).

While the mouse pointer is over any part of a menu (including the menu title or any of the menu borders), pressing the right mouse buttion will remove the current menu and return to the previous one. (Or will exit the program if this was the 'top level' menu.)

By placing the menu pointer on the horizontal border above the top menu item display, pressing the left button will cause the highlight bar to scroll through the list of options 'up-wards'. Likewise, if the mouse pointer is positioned at the horizontal menu border beneath the last menu option displayed, pressing the left button will cause the highlight bar to scroll through the list of options 'down-wards'.

The Paldef utility uses the mouse in the same way as PcMenu does.

How To Colour Your Menus

You colour your menus with the PalDef Utility. This utility will let you define up to fifty extra palettes. Palette 0 is 'Hard Wired' into the program and cannot be changed. This is the default palette you get when no palette number is specified.

To define a palette with the Paldef utility, simply move the bar onto the palette number you wish to define. To the right you can see the current definition for that palette in the Current Palette Definition Menu. The colour it displays is the existing colour definition for this palette. To change the definition press the Enter key (or click twice with the left mouse button when the pointer is over the palette you wish to change). You will then be asked whether you would like to change the Menu foreground or background colour or whether you want to change the colour of the select bar. By selecting either foreground or background and pressing the Enter key you will be presented with a list of valid colours for that option. Simply select the colour you want and hit the return key. The Example Menu then reflects the choice you have made. If you are satisfied with your changes simply press the Escape key to exit out of the menus. (If you are using the mouse you can exit the active menu by placing the mouse pointer over it and clicking once with the right button).

To save your palette definitions simply press the Escape key from the Palette list Menu and you will be prompted as to whether you would like to save your changes or abandon them.

NOTE

The PalDef Utility looks for the file containing the palette colour information called PCMU$PAL.DAT in the current directory, unless the DOS environment variable PCMENU_PALETTE is defined, in which case it will look in the directory specified by this variable. (See the section titled **'Installation and Implementation of PcMenu'** for more information).

If the file PCMU$PAL.DAT does not exist PcMenu, will create the file when loaded for the first time (as will the PalDef Utility). In this case, all palettes except palette 0 will be Black print on a White background. (This enables users of monochrome screens to set up a color scheme that works!!)

```
 PalDef Ver 2.0   Copyright (c) 1989-92 Brian Habel ALL RIGHTS RESERVED

                                                         Forground Colors
   Palettes           Color Selection For Menu
                                                           Black
   Palette 1            Menu   Forground                   Blue
   Palette 2            Menu   Background                  Green
   Palette 3            Menu   Select Bar                  Cyan
   Palette 4                                               Red
   Palette 5                                               Magenta
   Palette 6                                               Brown
   Palette 7                                               White
   Palette 8                                               Dark Grey
   Palette 9                                               Light Blue
   Palette 10                                              Light Green
   Palette 11        Current Palette Definition            Light Cyan
   Palette 12                                              Light Red
   Palette 13             Option One                       Light Magenta
   Palette 14             Option Selected                  Light Yellow
   Palette 15             Option Three                     Bright White

<Esc> Exit Window  ↑↓ Choose Item  ↵ Select Item Tue Sep 22,1992 12:14:15
```

Fig. 2 Screen Layout of the Paldef Utility

More On Batch File Programming

Normally, you can place any batch file command in the Menu definition file you like when defining a menu option. There are however some special characters that can be used to enhance your option.

PcMenu does support the use of @ variable characters as used in the NETWARE Menu utility. These characters allow you to ask the user for further information interactively, which will then be added to the batch file when executed. The syntax is as follows.

@[n]"*prompt text*"

where n is the repeater or variable number of this prompt. If no variable number is supplied then the user supplied information cannot be reused later within the batch file definition.

In the following example a menu option has been created to copy files

```
COPY A FILE
        copy @1"Enter source file name" @2"Enter destination"
        echo @1 copied to @2
```

The first line uses the DOS copy command. The first @ command asks the user for the pathname of the source file and the second for the destination.

To fully understand how PcMenu Implements @ variables, you are encouraged to experiment a little and to look at the runtime batchfile that is generated.

PcMenu also supports the use of < > symbols for retrieving values from the DOS environment. You may use these in menu title names and menu option names. You cannot use them when writing Batch file commands, instead use the standard DOS Batch file % characters instead.

For example suppose you have a choice between two printers, one called Document and the other called Draft. If you set environment variables with the DOS SET command as follows

```
SET OTHER_PRINTER=Document
SET CURR_PRINTER=Draft
```

then you can have a Menu Option defined as

```
Change printer to <OTHER_PRINTER>
    capture q=%curr_printer%
    set tmp=%other_printer%
    set other_printer=%curr_printer%
    set curr_printer=%tmp%
```

Then as this option is selected it will 'swap' the values of the two DOS environment variables and also the menu option will display the unselected printer.

NOTE
With DOS you may use the @ character to suppress statements from being displayed and the < > symbols for file redirection. With PcMenu the @, <, >, characters have special meanings. In PcMenu the < > characters are use to define DOS environment variables for Menu Titles and Menu Options. These characters retain there file redirection function in the DOS BATCH definition area of the Menu Definition file. The @ character, on the other hand, cannot be used a the DOS statement suppression character, as it has a special meaning to PcMenu throughout the entire Menu Definition file.

There is little reason to use @ characters as statement suppresses as they can be used in the startup batch file in conjunction with the bathfile 'echo off' command. (See the section below entitled 'How it all works')

PcMenu's Information Box

A nice feature of PcMenu is the Information Box. If you create a file called PCMU$HDR.DAT and place text in it, It will be displayed at the top of the screen inside the Copyright Information Box. This allows System Administrators of Networks to Display Important messages to users in a moments notice. Also you can tell PcMenu to get different Header Information files for different users by setting the a DOS environment variable called PCMENU_HEADER to the <u>directory</u> in which PCMU$HDR.DAT is located for that user.

You can change the colour of the Information Box by a series of colour codes.

To change the colour of the Information Box from its default the first line of the file PCMU$HDR.DAT must contain a combination of the following:-)

$bn	-	To change the background colour to colour n.
$fn	-	To change the foreground colour to colour n.
$sn	-	To change the borders of the Information Box to colour n.

(Where n is a digit corresponding to its assigned colour as defined in the table blow)

This then in effect sets up a new set of colour defaults.

To change the colour during a line of text you can use the following :-)

@fn	-	To change the foreground colour to colour n.
@bn	-	To change the background colour to colour n.

(Where n is a digit corresponding to its assigned colour as defined in the table blow)

This enables you to change the colour of text 'on the fly', so to speak, which will allow you to create colourful message!

NOTE
Background colour numbers must be between the range of $0 \le n \ge 7$ whereas, foreground colours must be within the range of $0 \le n \ge 15$.

Colours numbers are defined as follows

Information Box Colour Numbers	
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light Gray
8	Dark Gray
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	White

Fig 3. Colour Table

Text is automatically centred unless quoted with double quotes, in which case, text will be left justified and placement of words can be calculated manually.

You can also use the < > symbols to place the value of any DOS environment variable within a line as well.

Example
The following is an example of a PcMenu Information Box Definition file (PCMU$HDR.DAT).

```
$b1$f15$s14                             -
@f2H@f3e@f4l@f5l@f6o @f2T@f3h@f4e@f5r@f6e
```

(The first line sets up the Information Box with a Blue background, white foreground, with a yellow border. The second line will print the string 'Hello There' in a multitude of colours - a rainbow effect if you like)

IMPORTANT

When deciding on the location of Menus you count the top of the screen as the first line immediately below the Information Box!

Installation and Implementation of PcMenu

To install PcMenu choose create a new directory (preferably) on your hard disk (or if running a network on your servers hard disk). For illustration purposes we will choose the directory name of C:\PCMENU.

Copy PCMENU.EXE, PALDEF.EXE and MENU.BAT in this directory and place this directory in your path.

PcMenu uses a series of DOS Environment Variables to tell it where its files are. These are as follows:-

MENU	- Used only by MENU.BAT and defines the directory and filename of the PcMenu Definition File.
PCMENU_PALETTE	- Defines the Directory where Paldef and PcMenu is to find the colour palette file PCMU$PAL.DAT
PCMENU_TEMP	- Defines the Directory where PcMenu finds its temporary file when reloading after executing a DOS batch file. The file is called PCMU$TMP.DAT
PCMENU_HEADER	- Defines the Directory where PcMenu looks for the Information Box definition file PCMU$HDR.DAT
PCMENU_BATCH	- Defines the Directory where PcMenu writes the DOS Batch file for the selected option. The MENU.BAT batch file also uses this DOS Environment Variable.

DOS Environment Variables can be set in the AUTOEXEC.BAT file or from login scripts within a network environment.

If any of the environment variables are not set then PcMenu (and PalDef for the PCMENU_PALETTE variable) will look for files in the current directory, the instant they require it! It is recommended that all variables be set.

How it all works

PCMENU.EXE is not usually invoke from the command line, rather the entire system is driven from a DOS batch file called MENU.BAT of which a listing is made below:-

```
@echo off
pcmenu %MENU% /s300
if errorlevel 1 goto error
%PCMENU_BATCH%\PCMU$BAT.BAT
:error
```

This file will load PcMenu with the Menu Definition File located in the path defined in MENU. In this example PcMenu has been invoked with a screen saver time value of 5 minutes.

When an option is selected from a menu, and that option contained DOS Batch file code (ie. not a sub-menu), PcMenu will write the batch file code to the file PCMU$BAT.BAT (in the directory defined by PCMENU_BATCH) and record which menu and option that was selected by creating a temporary file in the directory defined by the DOS environment variable PCMENU_TEMP. PcMenu then exits back to MENU.BAT where the next instruction will execute the newly created batch file and hence the option just selected. After completion, MENU.BAT is re-executed (this is done from PCMU$BAT.BAT), and PcMenu is reloaded, and upon detecting the presence of the temporary file created earlier, will restore its state to the instant before the option was selected.