PcMenu

Version 3.0

User Manual

Table of Contents

Legal Stuff

Introduction

Installation of PcMenu

How To Use PcMenu

The PalDef Colour Program

Legal Stuff

Grant of Rights

PcMenu and PalDef are Copyright © 1989, 1990, 1991, 1992, 1993 by Brian Habel.

Although subject to copyright, this is **FREE** software providing the restrictions below are adhered too.

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

Restrictions

You may **not sell** this software.

You may use this software anywhere any place where NO personal financial gain is procured as a direct result of its use.

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 (and are indeed encouraged to) 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.

NOTICE

NOVELL and NETWARE are registered trademarks of Novell Inc., Provo, Utah, USA.
WordPerfect is a registered trademark of the WordPerfect Corporation in USA, UK, and in other countries.

Registration

If you use this software either in your place of employment, or on your home PC, the Author would like to know. At the back of this User Guide you will find a Registration Form asking a few simple questions about how and where PcMenu is used. Registration Forms can be sent by post to the address below. Users who have Internet access can use the file supplied and e-mail it to 'pcmenu@arcadia.cs.rmit.EDU.AU'.

You are under no obligation to do so. However, if you do, the Author may, from time to time, inform you as to the availability of updates to this software.

Although PcMenu is available to you for no charge, the Author has spent considerable time, effort, and money on this project. If you feel he should be further rewarded, please feel free to send any monetary donation you feel appropriate to the author, at the address below.

Credits

This software was written and designed by Brian Habel, a computer programmer and system administrator at the Bundoora Campus of the Royal Melbourne Institute of Technology.

Special mention is made of Michael Bernardi [mike@childsoc.demon.co.uk] from the Children's Society, London, England. Without his constructive comments, and amendments to this document, this release would not have been possible. Also thanks to Ivan Cekalovic, Frank Manzi, and the rest of the other Beta testers from RMIT. (Thanks Guys !)

Lastly thanks to all the others, who made constructive comments and suggestions about PcMenu Version 2.1 Your input to this release has been invaluable.

Contacting the Author

Those people who would like to make suggestions, comments or bug reports can contact the author (for those lucky enough to have Internet access) by e-mail to 'pcmenu@arcadia.cs.rmit.EDU.AU'.

Those without Internet access can reply by post to the following address:-

Brian Habel
c/o PO BOX 518
Heathmont 3135
Victoria, Australia.

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 their system so that it can be easily accessed.

This is achieved by creating customised screen menu's from a series of menu scripts, which 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 is NOVELL compatible, it has even more to offer, such as 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 and define 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) programs with PcMenu.

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 a plethora of batch files roaming your system as all files can be integrated into an organised group of files or if preferred, just one !

Installation of PcMenu

System Requirements

PcMenu will run on an IBM PC, XT, AT, or true 100% compatible, with at least 256k of free memory (the more memory available the more menus can be displayed at the same time), with either a MDA, CGA, EGA, VGA display. PcMenu is supported, and well suited, to run in a Network environment, it has been tested extensively on Novell Netware v3.11, but should work equally as well on most other platforms .

Copying the files to your System

Installation on a standalone PC

PcMenu is distributed in ZIP format. You will need to have an appropriate unarchiving utility that can handle this format. Configuration files and example menu definition files are designed to work from the C:\PCMENU directory. It is recommended that newcomers to PcMenu install the software this way first, if possible.

1. Extract the files into C:\PCMENU with the following DOS commands

```
cd  c:
md c:\pcmenu
cd c:\pcmenu
unzip a:\pcmenu30.zip
```

Substitute a: in the last command with the drive letter (and directory if appropriate) containing the pcmenu.zip file if necessary

2. PcMenu is ready to run, the Example Menus supplied. To load, simply run the demo.bat[1] file supplied by typing :-

```
demo <Enter>
```

You are now recommended to read the next section '**How it all works**' followed by the section '**Setting up the ini file**'.

[1] PcMenu uses a batch file called 'DEMO.BAT' to run the supplied demo menu files, however this name can be changed to any batch filename you prefer. Hereafter this file will be referred to as simply 'MENU.BAT'

How it all works

PCMENU.EXE is not usually invoked 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
rem
rem  Place this batch file in your PATH
rem  Setup a environment variable specifying where the Menu script
rem  is in %menu% before running this batch file.
rem  This can be done here, but is best done through Network Login
rem  Scripts if running on a network.
rem
rem  Only set, if not already set.
set pcmenu_ini=c:\pcmenu
if %menu%+==+ set menu=c:\pcmenu\brian.mnu
if %othermenu%+==+ set othermenu=c:\pcmenu\brian2.mnu
rem Some programs muck screens up, to fix run mode command
mode co80
rem mode mono
pcmenu %menu%
if errorlevel 1 goto exit
rem
rem Using clever arrangement of environment variables
rem you can set up exactly where you want pcmu$bat.bat file
rem to be written.
rem  ie.  f:%home%\pcmu$%stn%.bat
rem  Also you can setup a unique var ( ie %stn% ) for each workstation
rem on a Network, then each user has his own batch file generated.
rem
rem The default is setup below
rem
cls
c:\pcmenu\pcmu$bat.bat
:exit
```

This file will load PcMenu with the Menu Definition File located in the path defined in the DOS environment variable MENU. The demonstration menus have the screen saver set to a time value of 5 minutes in the Ini file.

When an option is selected from a menu, and that option contains 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.

Upgrading from PcMenu Version 2.1

PcMenu Version 3.0 is fully downward compatible with Version 2.1. Hence, all that is required to upgrade to version 3.0 is to replace the files PCMENU.EXE and PALDEF.EXE from version 2.1 with the ones from version 3.0. All DOS environment variables (as explained below) used in the earlier release are still supported. However, users are strongly encouraged to use the ini files to enable the new features of PcMenu. See the section below and the section entitled '**Setting up the INI file**'.

Users of Version 2.1 are also recommended to read Appendix A. '**Compatibility between Version 3.0 and Version 2.1**'

Implementation

Users who are installing PcMenu for the first time are recommended using the 'ini' file mechanism and skip to the section entitled '**Setting up the INI file**'.

The following is intended for those Upgrading from Version 2.1

PcMenu uses two mechanisms to tell it where its files are:-

1). The first way is to use the Version 2.1 DOS Environment Variables.

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 are 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, (or in the 'ini' file if used , see below) the instant they require it! It is recommended that if this approach is used that all variables are set.

2) The second approach, and preferred (especially if running on a Network) is to use PcMenu's 'ini' file. For details on its use see the next section below entitled '**Setting up the INI file**'

Setting up the INI file

New in PcMenu Version 3.0 is the 'initialisation' file, more easily referred to as the 'ini' file. The ini filename used by PcMenu is PCMU$INI.DAT and is located by the program in two ways. (As PalDef uses this file as well, to locate the directory the palette file is located in, it uses these same mechanisms as well)

1) By using the DOS Environment Variable PCMENU_INI, which is set to the DIRECTORY containing the ini file.

or

2) by using the "/I path " option on the command line (of either PCMENU.EXE or PALDEF.EXE). Where "path" is the DIRECTORY containing the ini file.

The ini file contains options to tell PcMenu where to locate files that it needs to use during operation. It is also used for setting up new features introduced to PcMenu Version 3.0.

INI File Options

The ini file is just an ordinary text file containing lines consisting of options and their associated values. Also you can use the '#'character to denote a comment line to help remind you of what each option is for, and for other incidental information. Any text appearing after a '#' character is ignored by PcMenu.

It is also possible to supply information from the DOS environment to any of the options in part or whole by using the < > characters. Simply place the name of the DOS environment variable inside the < > characters (eg. <COMSPEC> will be replaced by the path of where COMMAND.COM is located).

The ini file is hopefully self explanatory (a listing of an example ini file is located in the Appendix), however some options perhaps require further explanation.

MENU_BATCH

As explained previously in the section '**How it all works**' PcMenu is driven from a DOS batch file. By default the batch file name that PcMenu uses is MENU.BAT. Users may prefer another name (eg. PCMENU.BAT) to distinguish it from other menu systems currently installed . PcMenu uses this value to correctly (re)call the batch file that will enable itself to reload, after an option has been selected and completed.

BATCH_FILENAME

This option specifies an alternate filename for the PCMU$BAT.BAT. This is of primary importance if running on a network, particularly when the same user logs on twice with the same account. Most networks have the ability to distinguish which workstation a user is using, and by employing this information by storing it in a DOS environment variable.[2] This enables PcMenu to create different file names for the same network account, allowing the menu system to operate correctly for each user.

For an illustration of how this is implemented see Appendix C **The format of the initialisation file & use with NetWare** for an example on a Novell network.

INFO_BOX_UPDATE

Network System Administrators may, from time to time, change the contents of the Information Box. In PcMenu V.1, the Information Box was only updated when the System was (re)loaded. However, with this option you can have PcMenu check the PCMU$HDR.DAT file to see if it has been modified. It does this by simply checking to see if the date and time stamp on the file has changed since the last time it was modified. The value supplied to this option is the time (in seconds) that you want PcMenu to check the time stamp. Note however, that with smaller values, more load will be placed on the server. As to the correct value, one will have to experiment.

If a value of zero is specified then no time checking is performed and PcMenu will only update the Information Box when reloaded. (As was the case with PcMenu Version 2.1)

SCREEN_SAVER

As an alternative to the /s command line parameter you can place the value in seconds of inactivity you want the screen saver to activate.

A value of zero will disable the screen saver

[2] Check with your networking system documentation as to how you can store values of workstation ID's into the DOS environment.

ALPHA_KEYING

PcMenu now supports "Alphanumeric Keying". In addition to using the arrow keys users may type the option title they want to use. Only enough characters to uniquely qualify the menu option is necessary. Once the option required is highlighted you can select (execute) it by pressing the Enter key.

TABS

The TABS option allows System Administrators to easily format Menu options in columns. Each value separated by commas, is the number of spaces from the left edge of the menu you want to start the text from. To have PcMenu 'tabulate' insert true tab characters into the text, where each succeeding tab character tabulates to the next column specified in the TABS list. If there are more tab characters in a line than values specified in the TABS list they are ignored.

HELP_BAR_COLOURS

This option allows the option of setting the colour of the Help Bar located at the bottom of the screen to follow the global colours specified for the Information Box, or always retaining the system default colours.

SAVER_MESSAGES

The screen saver now displays messages across the screen every few seconds. There are two built into the system with an optional third which is user defined (see below). This option allows the user either to enable or to disable all screen saver messages.

SCREEN_SAVER_MESSAGE

Useful messages can be displayed as part of the screen saver with this option. Every third message that is displayed contains the text assigned to this option. The length of this message cannot exceed 80 characters.

Note that text must be enclosed in quotation (") characters, otherwise text will not be displayed as intended.

How To Use PcMenu

Planning Your Menus

Some thought must be given to 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 put 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 themselves, 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 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[,filename]
                  |
     Dos Batch file commands
                  |
                  OR
     %sub-menu_title[,alternative_filename]
```

Looking at this in more detail, each menu definition starts with the % character which must appear in the first column of the file, 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 Menu
Word Processors
        %Word Processors Menu
Dos Command
        echo off
        @"Enter your DOS command"
Other Options
        %Submenu File,C:\PCMENU\PCM_SUB.MNU
%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 Menu') and one sub-menu (entitled 'Word Processors Menu'). The main menu uses default values for menu location and palette colour. The sub-menu uses the default menu location but uses the user defined palette colour 2.

The 'Other Options' is actually held in a file called PCM_SUB.MNU, with a title of %Submenu File.

The main menu would display something like the following.

Brian's Menu
Word Processors
Dos Command
Other Options

Fig 1. Example Menu 'Brian.mnu'

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 that they appear in the menu definition file.

Option Hiding

After the menu option, you can add an optional filename path separated by a comma. PcMenu will check that this file exists before displaying and making that option available to the user. Network Administrators can (and usually do) make certain directories available to selected users. This allows administrators to reveal options to a select group of users using the same menu definition file, by simply specifying a file located in the restricted directory after an option.

For an Illustration on how option hiding is implemented, follow the following example (Using the previous Menu Script)

```
%Brian's Menu
Word Processors
        %Word Processors Menu
Dos Command
        echo off
        @"Enter your DOS command"
Other Options,C:\CONFIG.SYS
        %Submenu File,C:\PCMENU\PCM_SUB.MNU
%Word Processors Menu,,,2
Word Perfect 5.1
        echo off
        c:\wp51
        wp
        cd c:\
PC  Write,C:\DOS\FORMAT.COM
        echo off
        cd c:\pcwrite
        ed
        cd c:\
```

In this example, the main menu option 'Other Options' has a filename 'C:\CONFIG.SYS'. For this option to be displayed the file C:\CONFIG.SYS must exist and be readable. (Also, since the option calls for an Alternate sub-menu the alternate file must also exist, see the next section below)[3].

There is also a filename specified after the 'PC Write' option in the 'Word Processors Menu', again for this option to be displayed to the user, the File C:\DOS\FORMAT.COM must exist ! What filename you use is unimportant, it need have no relevance to the commands specified for an option. The filename is purely a simple mechanism to either display the option to or hide the option from a particular user.

[3] PcMenu checks for Option Hiding prior to the existence of an Alternate Sub Menu file. You can still use the option hiding technique, for options that call Alternate Sub Menu Files. In this case both the Option 'filename' and the Alternate sub-menu 'filename' must exist for that option to appear in the menu!

Alternate Sub Menu Files

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 can't 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.

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

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 plane (horizontal / vertical) .

Using Your Mouse With PcMenu

PcMenu supports any Microsoft Compatible mouse connected 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 button will remove the current menu and return to the previous one. (Or will exit the program if this was the 'top level' menu.)

After 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.

More On Batch File Programming

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

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 supports the use of the '!' character in the following manner. Any DOS batchfile command that starts with the '!' character will cause PcMenu to exit to DOS and <u>not reload</u> AFTER this command has executed. Hence, on a Novell Network you could have an option in a menufile...

```
Logout from Network
        !Logout
```

... and after the Logout command has executed the PC will remain at the DOS prompt.

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 used to define DOS environment variables for Menu Titles and Menu Options. These characters retain their file redirection function in the DOS BATCH definition area of the Menu Definition file. The @ character, on the other hand, cannot be used as 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 suppressors as they can be used in the start up batch file in conjunction with the batch file 'echo off' command. (See the section above 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$HDR.DAT and place text in it, It will be displayed at the top of the screen inside the Copyright Information Box. This allows System Administrators of Networks to Display Important messages to users at a moments notice. You can tell also PcMenu to get different Header Information files for different users by setting the DOS environment variable called PCMENU_HEADER to the directory in which PCMU$HDR.DAT is located for that user. Alternatively, you can set the location of the directory in the 'ini' file! See the section '**Setting Up the INI file**'.

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:-

$b - To change the background colour to colour n.

$f - To change the foreground colour to colour n.

$s - 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 :-

@f - To change the foreground colour to colour n.

@b - 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 ≤ n ≥ 7 whereas, foreground colours must be within the range of 0 ≤ n ≥ 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 Grey
8	Dark Grey
9	Light Blue
10	Light Green
11	Light Cyan
12	Light Red
13	Light Magenta
14	Yellow
15	White

Fig 2. 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)

The size of the Information box is limited to 15 lines to prevent forcing Menus off the bottom of the screen, although anything over 5 lines begins to look unaesthetic ! Organise your displays with some thought of this in mind.

IMPORTANT

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

The PalDef Colour Program

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.

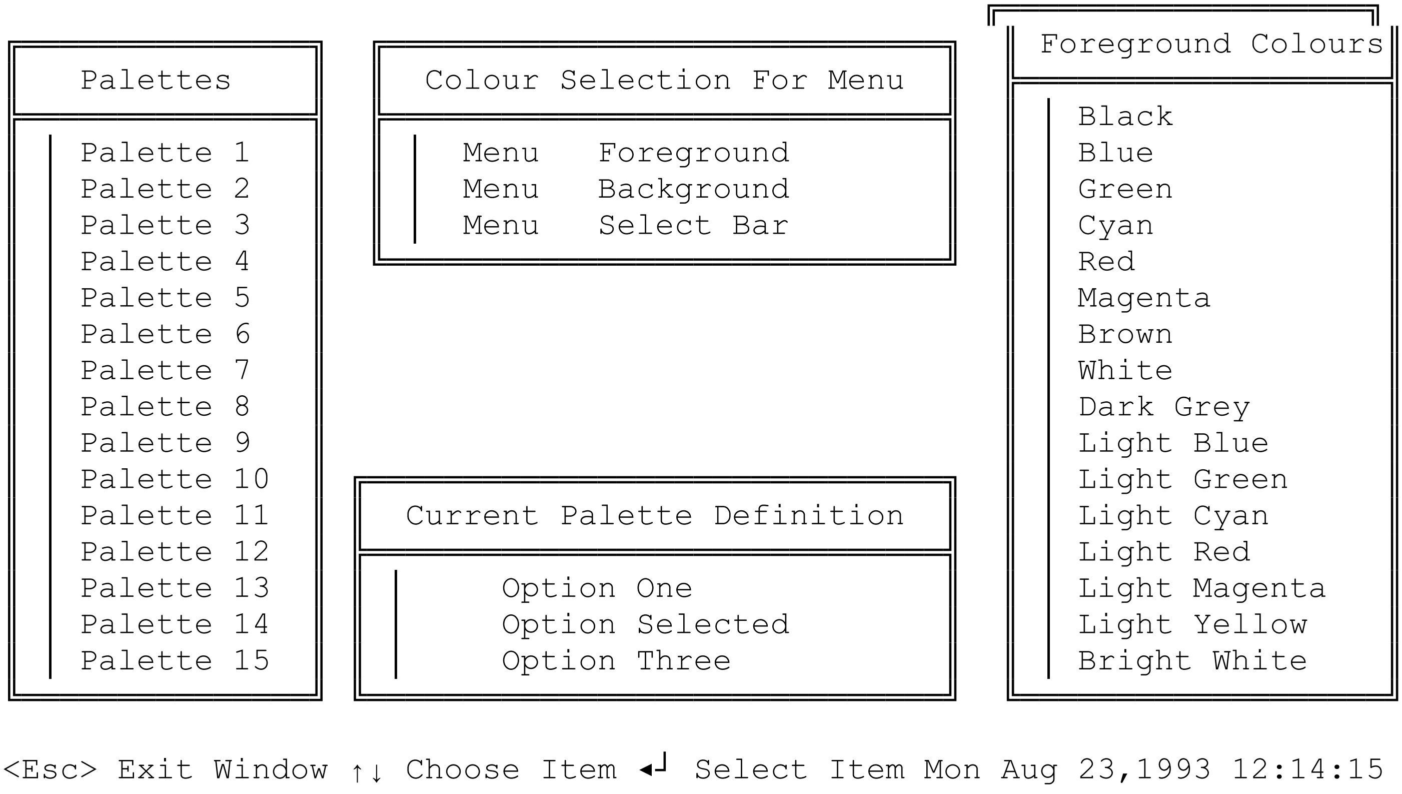
NOTE

The PalDef Utility looks for the file containing the palette colour information called PCMU$PAL.DAT in the current directory. If a directory has been assigned to the 'palette' option in the ini file then it will look in the directory specified by this variable unless the DOS environment variable PCMENU_PALETTE is defined, in which case it will look there! (DOS Environment Variables override the values specified in the 'ini' file.)

See the section titled **'Installation 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 colour scheme that works!!)

```
PalDef Ver 3.0   Copyright (c) 1989-93 Brian Habel ALL RIGHTS RESERVED
```



```
Palettes            Colour Selection For Menu         Foreground Colours

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

<Esc> Exit Window ↑↓ Choose Item ◄┘ Select Item Mon Aug 23,1993 12:14:15
```

Fig. 3 Screen Layout of the Paldef Utility

Appendix A

Compatibility between Version 3.0 and Version 2.1

A couple of problems I've noticed between this version and 2.1

- Old PCMU$LOG.DAT files are not compatible with this version This usually causes PcMenu not to find its "menu" file. If this happens, simply delete the 'LOG' file and try to start the menu system again!

- The INI file can cause confusion at times. If you want to use the values in the INITIALISATION file then, to avoid confusion, make sure that all DOS Environment Variables, PCMENU_BATCH, PCMENU_TEMP, PCMENU_HEADER, PCMENU_PALETTE are "Undefined". You have to use the DOS Environment Variable PCMENU_INI or the /I option on the command line to tell PcMenu (and PalDef) the DIRECTORY to look for the INI file.

A list of all the changes made to this version and those not found in this documentation are found in the file HISTORY.TXT.

Appendix B

Amendments and Additions to PcMenu Version 2.1

A lot of the changes to PcMenu Version 3.0 have culminated from those people who responded to my initial release of PcMenu

To those people, Many thanks ! (And now for your reward)

Changes/Additions in this version include

- An initialisation file (called PCMU$INI.DAT) as an alternative to using valuable DOS environment space. See the example file included with this release, it should be self explanatory !

- Periodic update of the information Box. Whenever a change is made to the file PCMU$HDR.DAT, PcMenu will detect the change (by use of the DOS file time stamp), reload and display the contents of the file. So, even if PcMenu is running passively, the Information Box will be updated immediately. It will also re-display the screen if the screen is blanked.

- Dos Environment expansion now works correctly for Menu Titles !

- Alt-F10 "Exit to DOS" key sequence has been added. Allowing for immediate exit to DOS regardless of what menu is displayed !

- Alpha Keying has been added !

- The Novell's " !logout " command now works !

- Display location problems have been fixed. Menu shadow's no longer wrap around as they incorrectly did previously.

- Menu option wrap around has been removed in this Version as it truly was unrealistic !

- Type ahead buffers are no longer flushed !

- Improvements have been made to the error reporting code.

- PcMenu now cleans up the batch files it creates.
  Users of DOS 3.20 may see superfluous batch file code as they ESCape from PcMenu. Later DOS versions appear to behave as intended.

- A bug in the Date and Time display has been fixed.

- External file Sub Menu bug has been fixed.
  Sub menus from external files did not load correctly under certain circumstances.

- Buffer overruns, that sometimes occurred when the last line in menu scripts are not terminated with a RETURN character have been fixed.

- The screen saver now optionally displays a simple message across the bottom of the screen. This is to differentiate between a computer that has been switched off and one running this screen saver !

- It was felt that when Information Box colours were changed, that the help bar colours clashed ! The option has now been added to the PCMU$INI.DAT file so that when global colours are set for the Information Box they also apply to the help bar as well.

- Added clipping to the PCMU$HDR file. The file can only contain 14 lines of information (excluding global colour info if present).

- Added the 'Option Hiding' Feature !

Appendix C

The format of the initialisation file & use with NetWare

This example is configured for use in the Novell NetWare environment. The PcMenu files are held in the sub-directory M:\APPS\PCMENU. This allows BOTH PcMenu and Novell's original Menu program to be used on the same network. The program runs from the BATch file PCMENU.BAT (see below), as opposed to PcMenu's default MENU.BAT, to avoid conflicting with Novell's MENU program. Add ROS rights to the group EVERYONE for this new directory. PCMENU.BAT should be copied to a mapped search directory, eg SYS:PUBLIC

Add to the system login script

DOS SET STN="%STATION"
DOS SET MENU="MENU.MNU" If this is set the command PCMENU.BAT on its own will run this default menu.
DOS SET NAME="%LOGIN_NAME"

PCMENU.BAT

```
echo off
rem Some Displays get their current mode mucked up !
rem Reset it back to text mode
rem mode co80
rem Start up PcMenu
set pcmenu_ini=F:\APPS\PCMENU
cd M:\apps\pcmenu
M:
rem use parameter as menu name otherwise use variable %MENU%
pcmenu %1 %2 %3 %MENU%
cd M:\apps
H:
if errorlevel 1 goto exit
rem Use the %STN% variable or, if the same person logged into the network
rem twice, the menu wouldn't know which one to run!
H:\USERS\%NAME%\pcmu$%STN%.bat
:exit
cd f:\apps
H:
```

PCMU$INI.DAT

```
#
#  PcMenu Version 3.0.x  Initialisation file.
#  PalDef Version 3.0.x  also uses this file to locate Palette data file.
#
#  You can specify the directory where this file can be found using the
#  /I option on the command line ( again with both PcMenu and PalDef )
#
#  Alternatively, both these programs use the NEW Dos Environment Variable
#  PCMENU_INI when set to the directory containing this file.  ( Which will
#  override the /I option )
#
#  DOS ENVIRONMENT VARIABLES ALWAYS OVERRIDE THE VALUES IN THIS FILE !
#
#  As you have probably gathered by now the '#' character is a comment !
#
#  This file consists of 'parameters' = 'value'
#
#  'value' 's can include values from DOS ENV by surrounding the
#   DOS Environment Variable name in < > characters.
#
# PALETTE  specifies the directory in which the file PCMU$PAL.DAT
#         DOS environment variable PCMENU_PALETTE will override this
#         value.
#
palette  = M:\APPS\PCMENU
#
#
# HEADER   specifies the directory in which the file PCMU$HDR.DAT
#         DOS environment variable PCMENU_HEADER will override this
#         value.
#
header   = M:\APPS\PCMENU
#
#
# BATCH    specifies the DIRECTORY that you want the file PCMU$BAT.BAT
#         to be created.  This directory must be read/writable.
#         DOS environment variable PCMENU_BATCH will override this
#         value.
#
#         **Note the value placed here must also be placed in MENU.BAT**
#         ( or whatever you are using to control the menu system )
#
#
batch = H:\USERS\<NAME>
#
#
#
#
# TEMP     specifies the directory that PcMenu uses for its scratch files.
#         This directory must be read/writable.
#         DOS environment variable PCMENU_TEMP will override this
#         value.
#
temp  = H:\USERS\<NAME>
#
```

```
#
# MENU_BATCH   specifies the path and name of the controlling Batch file
#             ( or whatever ) that is used to reload PcMenu after
#             command launch.  ( Default: MENU.BAT )
#
menu_batch  = PCMENU.BAT
#
#
# BATCH_FILENAME  specifies the name of the batch file that PcMenu creates
#               in the directory specified by the 'batch' parameter or
#               the PCMENU_BATCH Dos Environment Variable.
#               ( Default: PCMU$BAT.BAT )
#
#         **Note the value placed here must also be placed in MENU.BAT**
#         ( or whatever you are using to control the menu system )
#
batch_filename = PCMU$<STN>.BAT
#
# INFO_BOX_UPDATE  where the 'value' is set to the number of seconds that
#                you want PcMenu to check the PCMU$HDR.DAT file for
#                modification.
#
#            ( Warning - Small Values may put heavy loads on File Servers )
#
#                if 'value' is set to 0 timed updating is not used,
#                and PcMenu only loads the Information Box on initial
#                startup and when returning from DOS after an executed
#                menu option !    ( Default: 0 )
#
info_box_update = 100
#
#
# SCREEN_SAVER    where 'value' is set to the time of inactivity until
#               the screen saver is activated.
#
#               If set to ZERO the screen saver is disabled !
#
screen_saver = 300
#
#
# ALPHA_KEYING   Specifies whether you want alpha keying enabled !
#              set 'value' to 'on'  for enabled
#                       or 'off' for disabled ( Default )
#
alpha_keying = on
#
#
```

```
# TABS    selects tabulation points used in formatting Menu layouts.
#        Measured from the left edge of the window, each successive
#        TAB character found will start the cursor at the next tab
#        position.
#
#        'value' is specified by a series of tab positions in increasing
#        order, separated by commas.
#
#        Tab positions are used in 'Menu Titles' and 'Menu Options' Only.
#        There is no default value for this parameter so if this option
#        is not specified or the tabulation list is exhausted successive
#        TAB characters will be ignored.
#
Tabs = 8,16,24,32
#
#
# HELP_BAR_COLOURS    Define the colours of the help bar by using global
#                colours from Information Box Details.
#
#           'value' of 'default' is the default value for this option.
#
#            set 'value' to 'default' to always retain system Default settings
#            or 'infobox' to have colours follow GLOBAL colour
#            settings in the PCMU$HDR.DAT file
#
help_bar_colours = infobox
#
#
# SAVER_MESSAGES  Enables or Disables ALL Screen Saver Messages
#
#           set 'value' to 'on' to enable default saver messages, and
#                    the 'screen_saver_message' if set.  ( Default )
#
#                    to 'off' for NO messages.  The screen remains
#                    blank.
#
saver_messages = on
#
#
# SCREEN_SAVER_MESSAGE  Display your own message while screen saver Active.
#
#              This adds to the default list of screen saver messages
#              a user defined string "value" that will display
#              every THIRD time.  ( ie.  after the two default messages )
#
#              Text MUST be enclosed in quotes and can be <= 80 chars !
#
screen_saver_message= "PcMenu V3.0  now has MORE features to offer !"
#
```

Appendix D

Registration Form

Although you are not obliged to register, by doing so you will be notified of any further updates to PcMenu if and when released.

Name:

Organisation:

Postal Address:____

City:_________________________ Zip/Post Code:_____

State:________________________ Country:____

Email Address (If you have internet access): _

Where is this software being used ? □ Personal PC □ Education/Training

□ Comercial Business □ Government Dept.

□ Other _____

What Network are you using ? □ Netware □ Banyan □ 3Com □ Lantastic

□ Other ____

Comment:___