

## **Contents**

**[Introduction](#)**

**What it is**

**[Installation](#)**

**Installation and removal**

**[Setting Up Menus](#)**

**Customizing your menu**

**[Qmenu Options](#)**

**Setting options**

**[The Quick Menu](#)**

**Built in menu**

**[How to...](#)**

**Quick reference**

**[Registration](#)**

**This IS shareware**

**[Revision History](#)**

**What's new?**

**[About SawBuck Systems](#)**

Click your mouse on the Copy button, or press <ALT-C> to add the highlighted menu line to the menu highlighted in the menu list box. This button is active only when the menu lines box has the input focus. The selected menu line will be added to the end of the destination menu. This effect is immediate and is not affected by the Save or Undo functions.

Click the Delete button with your mouse to delete the currently selected menu or menu line. There is no keyboard equivalent for this function (to prevent accidents; besides, alt-D is spoken for by the Move down function). Since deleting menus is immediate and permanent, Qmenu will ask you to confirm this direct. There is no such confirmation for menu lines as this can be undone. Also note that when you delete a menu, it will not remove any existing references to that menu.

Click this button with your mouse or press <ALT-E> to edit the currently selected menu or menu line. The arrows on the button point to the list that will be affected. Double clicking the menu line or pressing <ENTER> on the highlighted line will also invoke this function. The edit function will open a new dialog for either menus or menu lines. Note that you cannot change the name of or delete the "Main" menu. Also note that a menu having the name Startup has a special function when using Qmenu as your Windows shell.

Click the Insert button with your mouse or press ALT-I to create a new menu or menu line. The arrows on the button indicate which list will be affected. When inserting menu lines, the new line will be placed directly above the currently highlighted item. The menu list (on the left) is sorted so it really doesn't matter which one is highlighted when you select the insert function; the list will be resorted.

**Note** that simply creating a menu will not cause it to appear in your menus. You must first insert it into a menu line.

**Menu Lines:** This box lists all items in the currently selected menu. Select a line entry by highlighting with the cursor keys or clicking once with the mouse. Once highlighted, you can use any of the edit functions on the selected line. Double clicking a menu line, or highlighting it and pressing <ENTER> will bring up the Menu Entry dialog (same as selecting the edit function).

## Drag and Drop

The easiest way to add items to the Qmenu menu is to drag programs from File Manager and drop them onto the Qmenu icon. Obviously, for this to work, the Qmenu icon must be visible. If it is not visible, either Qmenu is not running, or it is hidden (see [Qmenu Options](#)). Once you have both Qmenu and file manager running, simply click the left mouse button on the program you want to add to Qmenu, and while holding the button down, position the mouse pointer over the Qmenu icon and release the button. Qmenu will popup a dialog box with some default Options which you can accept, change or reject.

**Add to menu:** A name assigned to one of the Menu?? sections. The list box displays the names used in your menus rather than the actual section headings. This gives you a better idea of where the item will appear. When Qmenu inserts the line, it will replace the title with the section heading, Menu1, Menu2, etc.

**Entry name:** This is the text that will appear in the menu. The & before the W in the above example, causes the W to be underlined in the menu indicating that W is the hotkey for this menu item. Pressing the hotkey causes the light bar to move to the next entry that is assigned that key. If only one entry is assigned to a particular hotkey, it will be immediately selected. Do not begin a menu name with an @, Qmenu reserves this symbol for [special functions](#).

**Execute:** The name of the program to run. If the program file is located along your PATH, it need not have the full path name (as in the example). This may also be a document file that has a program associated with it in File Manager (or registration database). For instance, an entry of *c:\windows\readme.wri* would invoke Windows Write with that file.

**Command line:** Parameters to pass to the program. In the example, we pass two file names to EW (an editor) which causes it to load those files when it starts up. If this were DOS, it would be the same as typing *c:\ew\ew win.ini system.ini* at the DOS command prompt. Valid parameters are entirely the discretion of the program you are running.

**Initial directory:** The directory to log onto before running the program. This will be the invoked programs default directory. This is the same as typing *cd xxxxx* at the DOS prompt with the exception that it can also changed the logged drive.

**Window:** This determines the window that the program will be run in. 99% of the time it will be normal allowing the invoked program to determine its window size (this has no effect on DOS programs). Other options are [min](#), [max](#), and [icon](#). Note that programs are not required to follow this directive (Qmenu itself, will disregard it), although most Windows programs will abide.

This list box shows your current menus. To select a menu for editing, highlight it with your cursor keys then press <ENTER> or double-click the desired menu name with your mouse. Note that if you have unsaved menu line changes in the box on the right, they will be automatically saved. To switch between the Menu list and menu lines list, press the <TAB> key or click (once) with the mouse anywhere inside the desired box. Only the top three buttons in the center group apply to menu names.



Menu item List

Lists the currently defined menus. The names displayed are those you provide when creating new menus. As such, a single menu may be referenced by more than one name. To select a menu, highlight it using the mouse or keyboard then press TAB or ENTER to switch to the menu item list. You can also switch to the item list by clicking it with your mouse.

## Changes from previous versions

If you are upgrading from a version of qmenu earlier than 1.10, note that we have changed the way that qmenu handles menu naming. Qmenu will automatically (and unconditionally) update your existing ini file, but there are two things you need to be aware of:

In previous versions, you added a menu reference in the form:

```
@Menu2=Name
```

In version 1.10 and up, we still use the @menuxx= form, but the name is ignored. Instead, the name is derived from an entry in the MenuNames section. When Qmenu updates your ini file, it will assign the last referenced name to the Menuxx section. It will also remove the old [Menus] section. If you plan to use your own editor to maintain your menus, you may want to keep the [MenuNames] section updated. Other than this, you should have no problem if you decide to edit your ini file manually. You will still be able to use the internal editor.

In version 1.12, we have modified the handling of the Startup menu. Rather than having a section name of Startup, it is now in the same manner as all other menus, MenuXX. It is now identified entirely by its name. Qmenu will automatically rename your existing Startup menu to Menu41.

## Introduction

Quick menu is a tiny program designed to make working in Windows a little easier and perhaps quicker. *Its primary purpose is to provide a customizable menu that you can popup at any time to run your favorite programs.*

Qmenu also provides the following convenience features:

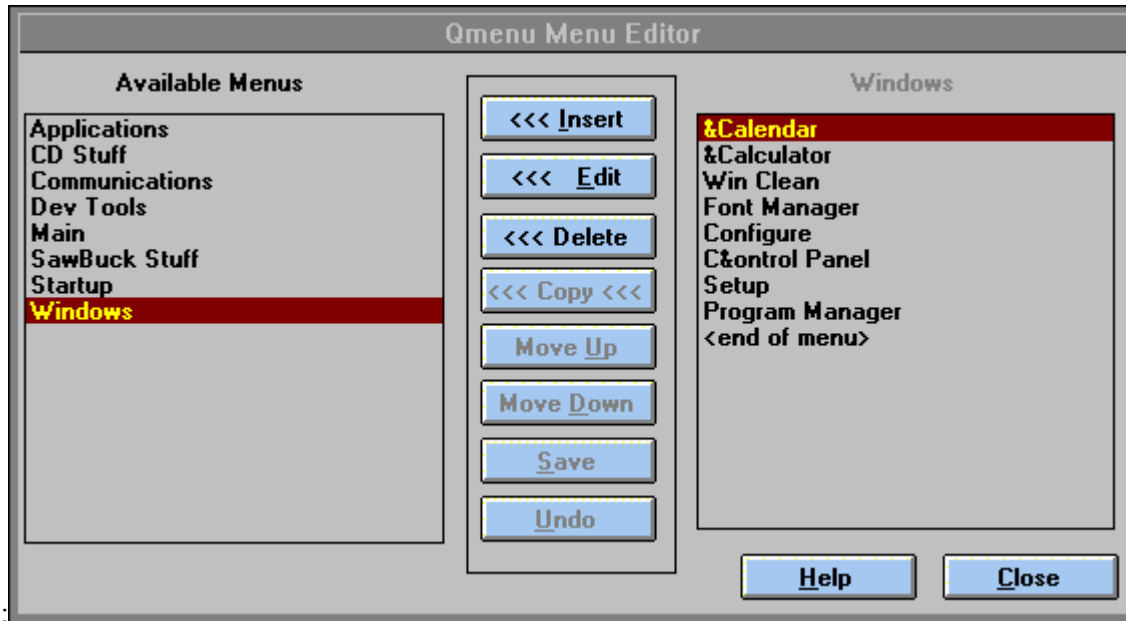
- Allows the [PrtScr](#) key to print the active window
- Quick exit from windows, also restart Windows, or reboot
- Quick access to help files
- Quick access to any program via a run dialog
- Quick form feed for your printer
- [Screen Capture](#) copies an area of the screen to the clipboard

By default, Qmenu will pop up by clicking the right mouse button on an inactive area of the screen. This includes the desktop (or backdrop) and any inactive windows. It does not include icons or the active window (the window that currently has the input focus). Qmenu can also be made to pop up with the middle button (on a 3 button mouse) or by pressing the scroll lock key. refer to: [Qmenu Options](#)

Qmenu is functionally similar to the many icon bar type programs that can be found on the market, in that it allows you to quickly access your favorite programs. Unlike the icon bars, Qmenu does not take up any screen real estate and uses very little resources (even less than program manager). If you have tried one these icon bars, then perhaps you know what I mean. The icons are always getting covered up, or in the way. What's more, some of those icons provide little clue as to what's underneath!

# Menu Editor

Here's a picture of the menu editor (yes, I know, it's quite large). Click an area of the picture for more information. More text follows:



The large list box on the left contains all your currently defined menus, while the list box on the far left shows the items in the currently selected menu. The <TAB> key will switch the input focus from one box to the other. Clicking the left mouse button within a box will also switch the focus. Since each list box always has a highlighted selection, Qmenu provides you with several other clues to identify which box currently has the focus. The first clue is the name appearing above each box. This name is shown in the disabled text color when the box does *not* have the input focus.

A second clue is in the group of buttons between the two list boxes. The top three buttons perform their functions on both menus and menu lines. Which they affect is indicated by the "<<<" and ">>>" arrows on the button. When the Available Menus box has the focus, these arrows point toward the left "<<<" (as in the screen above). When the menu lines box has the focus, the arrows point to the right ">>>". Furthermore, the copy and move buttons are active only when the menu lines have the input focus.

## Using the Menu Editor

The interface may feel a bit awkward at first, but I think you'll get use to it. The best way to do this is to just play with it. The only thing affected by the editor is your QMENU.INI file. It does not delete or change any other files on your disk. If you like, make a backup copy of QMENU.INI before you start experimenting. Then if it gets completely mangled, you can simply copy your backup over QMENU.INI.

## Modifying Menus

The menu editor allows you to change menu names, add new menus, and delete menus. To begin, set the input focus to the box on the left. Pay attention to the arrows on the function buttons, if they are pointing to the menu list, they will act on the currently highlighted menu name. Once highlighted, you can use any of the available buttons to make your changes. There isn't much to do with a menu except give it a name or delete. Refer to the picture above for details.

Use the Insert function to create a new menu.

Use the Edit function to change a menu name.

Use the Delete to remove a menu (Note it does not remove references to the menu from the menu lines).

### **Modifying Menu Lines**

Things work pretty much the same on the right hand box. Highlight one of the menu lines and select a function. One major difference here is that changes are not saved immediately. See the discussion below about the edit buffer.

### **The menu lines buffer**

Changes to menus (the box on the left) take effect immediately. However changes to menu lines (on the right) are not saved until one of three things occur:

- 1 - you select another menu to edit
- 2 - you select the Save or Undo function
- 3 - you close the dialog (or exit Qmenu)

Menu line changes are written to a memory buffer. By default, Qmenu creates an 8K buffer which should suffice for most users. If the buffer fills up, Qmenu will display a messagebox stating "Edit Buffer full, increase buffer size" and will disallow further editing (until you save or undo your changes). If you get this message a lot, you may want to do just that. To do so, you will have to edit your QMENU.INI file. In the section named [Preferences], you'll see a line that begins

Edit Buffer=*size*

Where *size* can be a number between 1024 (1K) and 65535 (64K) with the default being 8192 (8K). Unless you have some very, very large menus, you shouldn't need to change this. Nonetheless, it's there if you need it. Note that this memory is allocated when you first open the menu editor (and free'd when you close it) so changing it while the editor dialog is on your screen will have no effect. Also, there is no practical reason to lower this setting. If your system can't spare 8K for a few minutes, you're not going to be running any more programs, anyway!

## Installation

Installation is simple and straightforward. Simply copy the distribution files **QMENU.EXE**, **QMENUD.DLL**, **QMENU.HLP**, and **SAWBUCK.DLL** to a directory on your hard drive. Any directory will do as long as *all the files are in the same* directory. The only exception is QMENUD.DLL, which must be in your Windows directory (where win.com is located) *if you are using Qmenu as your primary shell*. If you receive Qmenu as a ZIP archive, you can unzip it into the desired directory. Qmenu will create **QMENU.INI** in this same directory, if it does not already exist. If you have registered your copy of Qmenu and are installing a new release (not a new version), you do not need SAWBUCK.DLL or QMENU.HLP. Nonetheless, you may want to install the help file (QMENU.HLP) just to see what's new. Also note that on-line help will not be available without QMENU.HLP. SAWBUCK.DLL and QMENU.HLP are **required** for all unregistered versions of Qmenu.

If you plan to use Qmenu as your shell please read the topic [Qmenu, A Shell](#).

NOTE: if you are upgrading an existing copy of Qmenu, make sure Qmenu is not running when copying the files. Once done, you will probably want to add Qmenu to your Windows Startup group so that it runs whenever you start Windows. Simply drag qmenu.exe from file manager to the Startup group in program manager. You're ready to roll!

Alternatively, you can manually add Qmenu to a Program Manager group by pulling down the File menu and selecting New. Make sure Program Item is highlighted, then press the OK button. You can use the Browse button to locate QMENU.EXE. Any startup directory will do, although, logically, it should be the directory where qmenu.exe resides.

### Removal:

To uninstall Qmenu, simply delete the program manager icon and the following files:

- QMENU.EXE
- QMENU.INI
- QMENU.HLP
- QMENUD.DLL
- SAWBUCK.DLL

Qmenu does not modify any other files.

## Setting Up Menus

Qmenu provides three methods for adding menus. The simplest is the drag and drop method, by which you drag programs from the file manager onto the qmenu icon.

The second method is to use Qmenu's Menu Editor. This provides a dialog for adding, deleting and changing menus and menu items.

The third method is to edit the Qmenu.ini file. Currently, this is accomplished with the Windows notepad. Refer to [qmenu.ini](#) for more information on this.



## Qmenu Options

The options in this dialog allow you to modify Qmenu's behavior.

**Show Window:** Check this box if you want the Qmenu icon to remain visible (note: it must be visible to drag programs to it from file manager). Uncheck the box if you don't want Qmenu taking up screen real estate.

Note: The Quick menu has a toggle to Show/Hide Qmenu. This allows you to quickly display and hide the icon.

**Hotkey Enabled:** Check this box if you would like to popup the menu by pressing a key. When you click the mouse to invoke Qmenu, it will first check to see if the mouse cursor is positioned on an active window. If it is, Qmenu will ignore the click and pass it on to the active application. As such, there is no way to call up the menu when running an application or maximized. One way to overcome this is to use a hotkey to invoke the menu. Take note that the key selected will be unavailable to other applications. Clear this box to disable the hotkey.

**Key code:** The virtual keycode to pop up the menu. Currently, Qmenu does not support key combinations (ctrl, alt, shift) since most seem to prefer the mouse. Anyway, here are some potential keys:

ScrollLock - 145 (the default)

Caps Lock - 20

NumLock - 144

F1 thru F12 112 thru 123

Keypad: / - 147 \* - 106 - - 109  
+ - 107 5 - 12

On the Microsoft Natural Keyboard:

left Windows - 91

right Windows - 92

application - 93 (to the right of the right windows key)

**Enable PrtScr:** Check this box to enable the PrtScr key to print the active window. For more information, see [screen printing](#).

**Magnification:** This controls the amount of stretching that takes place when Qmenu does a screen (window) print. With no magnification, the printed image tends to be quite small. 2 x will approximately double the size. Fit to page will set the width to the page width and adjust the length to maintain the aspect ratio. Generally, 2x provides the best balance between image size and printing speed. Windows are printed as bitmaps which can become quite large depending on window size, resolution, and the number of colors in the image.

**Activate Button:** Choose which mouse button you want to use to pop up the menu. Check the disable box to temporarily disable Qmenu. The disable function will automatically enable Show Window, since this is the only way you will be able reactivate Qmenu (click on the Qmenu icon and choose Options).

## Screen Printing

Many users find this to be one of Qmenu's most desirable features. By default, when you press the PrtScr (Shift-PrtScr on some keyboards), Windows copies an image of the screen into the clipboard. Unfortunately, Clipboard provides no function to print its contents. There are ways to do this but they all require several steps.

When PrtScr is enabled in the [Options](#) dialog, Qmenu will take over the PrtScr function and send an image of the active window directly to your default printer. For instance, if you are reading this from the Windows Help program (and Qmenu is running, and PrtScr is enabled), pressing PrtScr now will copy the help window to your printer.

**NOTE:** this function will not work on all printers. In order to work, your printer must be capable of printing raster graphics and, depending on the size of the image, bitmaps larger than 64K. Fortunately, this includes MOST laser, inkjet, and dot matrix printers. It has proven to work on Canon BJ series printers, Epson LQ (and probably others), and even an old Okidata 192. Quite frankly, most of the work is done by Windows, so if you can print a graphic, you will most likely be able to use this feature.

The quality and speed of printing depend on a number of factors. First and foremost is the quality of your printer, lasers and ink jets work the best. Your default printer settings also play a major role in determining the output (set through Print Manager or Control Panel). To find the best balance of speed and quality, experiment with the dithering and intensity controls in the printer options dialog. Line art dithering gives the best image, but takes the longest to print. Also, the higher the resolution the better the image will look (and take longer to print). Please note that the settings you make in Print Manager or with Qmenu's [Printer Setup](#) function, will affect the output of **ALL** windows applications that use the default settings.

Finally, this feature was designed primarily to print windows containing text that cannot be printed in any other manner. Although it will print any window, it assumes that the window contains text that you will want to read. Since light characters on a dark background do not look good on paper (they also take longer to print), Qmenu attempts to determine if the image is light on dark or dark on light (this algorithm is still being refined). If Qmenu determines that the image is light on dark, it will invert the printed image, that is, what was dark on the screen, will be light on the paper and vice versa.

## The Quick Menu

This menu allows you to perform some routine tasks and configure Qmenu. You have the following choices:

Options	Brings up the <a href="#">Qmenu Options</a> dialog
Menus	Brings up the <a href="#">Menu Editor</a>
<a href="#">Show/Hide Qmenu</a>	Toggles display of the Qmenu icon
<a href="#">Unload Qmenu</a>	Removes Qmenu from memory (ends the task)
<a href="#">Capture</a>	Capture a portion of the screen to the clipboard
<a href="#">Eject Page</a>	Sends a form feed to the default printer
<a href="#">Printer Setup</a>	Change default printer settings
<a href="#">Run</a>	Run any program
<a href="#">Get Help</a>	Quick access to any Windows Help file
<a href="#">Exit Windows</a>	Quit Windows
<a href="#">Restart Windows</a>	Quit Windows and restart it
<a href="#">Reboot</a>	Quit Windows and warm boot the computer

**Unload Qmenu** This option removes Qmenu from memory. This is the same as selecting the close option from the Qmenu icon's system menu.

**Show/Hide Qmenu:** This selection (Hide Qmenu or Show Qmenu) toggles the visibility of the Qmenu icon. This allows you to quickly hide it (for esthetics) or show it so you can drag files onto the icon.

**Eject Page** Simply sends a form feed to your default printer. Useful if you have a printer (such as mine) that requires a form feed to tear off the last thing printed.

**Run:** This selection allows you to browse your drives and select a program to run. Very much like Program Managers File Run command. Qmenu will remember the last program you ran from this dialog (provided it is run from a fixed drive).

**Get Help:** This option allows to have a directory of help files that you can quickly view. This is especially helpful if you have Windows help files that aren't associated with a particular program. Help files are a good place to store all kinds of reference information.



**Exit Windows:** Allows a quick exit from Windows. Similar to the exit windows option on Program Managers system menu. Like the Program Manager, it will query all processes and will not exit if a process refuses to exit (as most DOS programs will do). Unlike Program Manager, it won't ask you twice.

**Restart Windows:** This will perform a normal Windows exit and will then restart Windows. This is a wise thing to do if you get a GPF. Also useful for applying changes that only take effect when Windows is started.

**Reboot:** Performs a normal Windows exit, then reboots your computer. Same as exiting Windows then pressing Ctrl-Alt-Del.

## Screen Capture

This function (located in the [Quick Menu](#)) lets you capture a rectangular area of the screen and copies it to the Windows clipboard. When selected, the cursor will change to a cross. Move this cursor to a place on the screen that you want to capture. Press and hold the left mouse button. While holding the button down, move the mouse so that the box displayed surrounds the area you want to capture, then release the button. The selected rectangle is now copied to the clipboard. From there, you can do with it what you will. Paste it into an app such as Word, Write, Paintbrush, or whatever. Use your imagination! The screen shots in this help file were captured with this function.

Credit where credit is due, much of the code for this function was gleaned from a sample app provided by Microsoft Corp.

## Qmenu.ini

Qmenu.ini contains all your Qmenu option settings as well as your custom menu(s). Qmenu will look for this file **only** in the directory where Qmenu.exe is located. This file has the following sections:

### [Info]

This section just contains information on setting up the menu lines. You can add any notes you wish to this section.

### [menu1]

This section contains the primary menu. Menu1 **must** always exist. Each line in this section is a menu entry. The [info] section contains a brief description of these entries. For detailed information see [Menu Entries](#).

### [MenuXX]

[Additional menus](#) will be Menu2, Menu3 ... up to Menu40.

### [MenuNames]

This section is maintained by the [Menu Editor](#). It is used to keep track of the existing menus and the names assigned to them. Entries are in the form:

```
Menu1=Main  
Menu2=&Windows  
Menu41=Startup
```

When Qmenu is building the menu and finds an @Menuxx= line, it will look in this section for the name of the menu. If the name is not found, Qmenu will add a line using the menu and number as the default name:

```
Menu2=Menu2
```

This is to insure that menus added via an editor appear in the internal editor's dialog box even if the name is omitted from this section. Care should be exercised when adding menus manually, the first part of the line **MUST** be the string Menu1 through Menu40. The menu editor does not validate these so erroneous entries could cause problems.

### [Preferences]

This section contains your option settings. Normally, you should not need to edit this section. With one exception, all options are maintained through the [Qmenu Options](#) dialog. The exception is the [Edit Buffer= setting](#).

### [Registration]

This section exists only if you have registered Qmenu. Do not modify the entries in this section. Doing so could render Qmenu unusable.

## Menu Entries

Each line in the [menu?] section consists of 5 parts (or parameters) in the format  
**x1=x2,x3,x4,x5**

**Example: Config &Win=c:\ew\ew.exe,system.ini,c:\windows,normal,**

This example would produce a menu entry named **Config Win** which when selected would:  
change directory to c:\windows  
run c:\ew\ew.exe system.ini

see also: [Setting Up Menus](#)

**Execute:** The name of the program to run. If the program file is located along your PATH, it need not have the full path name (as in the example). This may also be a document file that has a program associated with it in File Manager (or registration database). For instance, an entry of *c:\windows\readme.wri* would invoke Windows Write with that file.

**Command Line:** Parameters to pass to the program. In the example, we pass two file names to EW (an editor) which causes it to load those files when it starts up. If this were DOS, it would be the same as typing `c:\ew\ew win.ini system.ini` at the DOS command prompt. Valid parameters are entirely the discretion of the program you are running.



**Initial Directory:** The directory to log onto before running the program. This will be the invoked programs default directory. This is the same as typing `cd xxxxx` at the DOS prompt with the exception that it can also changed the logged drive.

**Window Style:** This determines the window that the program will be run in. 99% of the time it will be normal allowing the invoked program to determine its window size (this has no effect on DOS programs). Other options are [min](#), [max](#), and [icon](#). Note that programs are not required to follow this directive (Qmenu itself, will disregard it), although most Windows programs will abide.

**MINimized:** This requests the program to run as an icon. This is the same as *icon* except that the newly created icon will retain the input focus. I can't think of any real reason to use this, but it is there.

**MAXimized:** Requests the program to take up every bit of screen space available when it runs. May be useful on smaller screens.

**ICONized:** Requests the program to run as an icon. The input focus should return to the program that had it before this was run.

## Special Functions

The @ symbol in a menu name directs Qmenu to take special action when processing that entry. Its use will be expanded in future versions. Currently the following functions are available:

@bar= causes Qmenu to insert a line (separator) into the menu. Purely cosmetic, but can make the menu easier to read.

@EjectPage= same as Quick Menu [Eject Page](#)

@GetHelp= same as Quick Menu [Get Help](#)

@PrintSetup= same as Quick Menu [Printer Setup](#)

@Reboot= same as Quick Menu [Reboot](#)

@Run= same as Quick Menu [Run](#)

@WinExit= same as Quick Menu [Exit Windows](#)

@WinRestart= same as Quick Menu [Restart Windows](#)

@MenuXX=MenuName Where XX is a number from 1 to 40. This inserts a submenu into the current menu with the title MenuName. For details, see [Multiple Menus](#)

## Multiple Menus

Currently, up to 40 menus may be defined. If more than 40 menus exist, Qmenu will display the message "**Menu limit exceeded**" and refuse to create them. Menu names are limited to 39 characters. Submenus (menus other than the top-level menu) may appear in any menu and may appear in more than one menu.

One word of caution: Parent menus should NOT be placed in submenus as this would result in an endless **recursive** loop when Qmenu builds the menu. For instance, if Menu1 contains Menu2, then Menu2 must not contain Menu1. If Qmenu detects such recursion, it will replace the offending menu line with **<<recursive>>** which will be grayed and disabled.

Submenus are defined by adding a new [MenuXX] section to the Qmenu.ini file. XX must be a number between 2 and 40 inclusive. This number should not contain leading zeros. [Menu1] is reserved for the top level menu and must always exist. All others are optional. Here's an example:

```
[menu1]
&DOS Shell=dosprmt.pif,,c:\,norm
Print Manager=c:\windows\printman.exe,,,norm
@bar=
@menu2=Windows
&File Manager=WINFILE.EXE,,,normal
&Calculator=\CALC.EXE,,,normal
@WinExit=

[menu2]
&Setup=winsetup.exe,,,normal
&Colors=control,color,,norm
&Printers=control,printers,,norm
```

The above will create two menus. Menu2 will appear in Menu1 with the title "Windows". When selected, menu2 will popup beside Menu1.

FYI, the last two lines in the above menu will run control panel and automatically invoke the applet named, i.e., color, printer, date/time, etc. You can do the same with any control panel applet, just include the name that appears under the icon in control panel.

see also [Setting Up Menus](#), [Menu Entries](#)

## Qmenu, A Shell of a Program

You may, if you wish, use Qmenu as your Windows shell in place of program manager. The advantage is that Qmenu requires far fewer resources than program manager. The major disadvantage is that most installation programs will create program manager groups to run their product; they likely won't update Qmenu's menus. Of course, you can always run program manager from Qmenu, and adding new programs to Qmenu really isn't all that difficult. The choice is yours. If you decide to make Qmenu your shell, DO read the remainder of this topic.

Qmenu modifies its behavior in two ways when it is run as the shell:

1. When Qmenu first loads, it runs all the programs listed on the *run=* line in your WIN.INI file. Next it will run (as icons) all programs listed in the *load=* line of your WIN.INI file. Finally, it will run all programs listed in the menu that has the name Startup.
2. That being done, Qmenu will behave pretty much as it did before, except that it will ignore any requests to close it (such as from the Task manager). In addition, the Close selection is removed from the system menu and replaced with "Exit Windows". "Unload Qmenu" is removed from the Quick menu. In other words, the only way to exit Qmenu is to exit Windows.

### Setting it up:

**Note:** Qmenu should be unloaded before attempting to move it.

The first thing you need to do is move QMENU.DLL to your windows directory (usually c:\windows). Alternatively, you can move all the Qmenu\*. \* files to this directory, if you'd like to keep them together. (If using an unregistered copy of Qmenu, you will also need to keep SAWBUCK.DLL in the same directory as Qmenu.exe).

Next, edit the SYSTEM.INI file which should be in the windows directory and locate the line that reads

```
shell=progman.exe
```

change this to:

```
shell=qmenu.exe
```

If Qmenu is not in your Windows directory, you will need to include the full path name, such as:

```
shell=c:\winapps\qmenu.exe
```

That's all there is to it. Now restart Windows.



## Revision History

v 01.05 3/2/94 -- initial release.

v 01.06 -- no public release

New About box now shows available memory and system resources. This box is only seen in the registered version. In the shareware version, it is replaced by the shareware info screen.

[Special Functions](#) (Exit Windows, Restart, etc.) can now be added to any menu.

Qmenu will now display an error message if it cannot execute the selected application.

v 01.07 -- no public release

When printing the screen, Qmenu now converts the image to a device independent bitmap (DIB). While this is somewhat slower, it should provide better color (greyscale) rendition on printers whose palette is smaller than the video device. It should also provide compatibility with a wider range of output devices.

Qmenu now supports multiple nested menus. Up to 39 menus may be defined. For details see the section on [Special Functions](#).

v01.08 - 7/7/94

Added [screen capture](#) to capture portions of the screen to the clipboard.

When using Drag and Drop to add menus, Qmenu would append a trailing \ to the startup directory. As a result, the shell command would not change to this directory before running the program. This has been corrected. In addition Qmenu will check for this condition and correct it before calling the shell.

Added [Hide/Show Qmenu](#) to the quick menu.

v01.09 - 8/22/94

Qmenu may now be used as the [Windows shell](#) in place of program manager.

Fixed bug preventing programs being run from menus numbered 9 and above.

Added workaround for problem that prevented registration from "taking" without exiting and restarting Qmenu.

v01.10 - 10/2/94

Finally found that nasty bug that was messing with the registration and fixed it. Workaround removed.

Added a [dialog editor](#) for maintaining menus. It's still a little rough, but it's a whole lot easier than using notepad. Of course you can still use Notepad, or any other text editor. If you are upgrading, you may want to read the notes concerning [changes](#) in this version.

Qmenu will now trap (and render harmless) [recursion errors](#) in menus. No message is generated, but the menu entry will show up as <<recursive>> and will be disabled.

v01.11 - no public release

Qmenu will now run ALL programs in it's Startup group instead of just the first.

Added a Printer Setup dialog function. Also, all Qmenu internal functions (except

About, Unload, Hide and Help) can now be added to any menu.

v01.12 - 11/17/94

Modified the handling of the Startup Menu. In previous versions, the Startup menu could not be accessed from another menu. The startup menu is now like any other menu, in fact all you need do to make the programs in a Menu start automatically (when Qmenu is the shell) is name the menu Startup. Current users, see changes from previous versions.

Did some cleanup and optimizing. Qmenu now requires about 4K less memory than in version 1.10

Also cleaned up the help file and removed many of the screen shots. While they're nice to have, it seemed a bit ludicrous to have a help file 4 times the size of the program.

## Registration

Qmenu is a shareware offering from [SawBuck Systems](#). As such, you are hereby granted a license to evaluate this software for a period not to exceed thirty (30) days. If you continue to use it after this trial period, you are required to pay for it. No legal mumbo-jumbo, if you like it, please register, otherwise trash it. It's that simple, just send in the [registration form](#) along with the registration fee. In return, we'll send you a registration number that will get rid of the opening info screen. We'll also send you a copy of the [license agreement](#), free!

Shareware allows you to try software before you buy it. If you've ever purchased a software package only to find out it was garbage, then you understand the value of shareware. You save money by knowing what you're getting before you shell out your hard earned cash. Moreover, shareware is usually priced lower than commercial products. But it's up to you to keep the concept alive.

When you register Qmenu, you will receive (via mail) a registration card with a serial number to be entered into the program. To enter your serial number click on the REGISTER button in the opening screen (also accessible from the ABOUT selection in Qmenu.) This will open a dialog box in which you type your name and serial number. These must be entered **exactly** as they appear on your registration card. Once completed, the shareware notice will no longer appear when Qmenu is first started. You may, at this time, delete the SAWBUCK.DLL file and if you wish, the QMENU.HLP file. Note, however, that these files MUST be present to run an unregistered Qmenu.

Quantity discounts and site licensing available.

## About SawBuck Systems

SawBuck Systems

P.O.Box 1172

Manhattan, KS 66502-0012

voice/fax: (01) 913 539-0348 voice 6:00 pm to 10:00 Central Time (usually); FAX:  
anytime

E-mail: 74017,1630 on Compuserve.

Internet: 74017.1630@compuserve.com.

OK, it's a funny name. But that's our business, software for a sawbuck! Our goal is to provide quality software at an affordable price. The price is a sawbuck, or a multiple thereof; you determine the quality. Now to keep prices this low, we obviously have to cut costs somewhere.

When you register SawBuck software, we'll send you a letter with your serial number and a copy of the license agreement. And that's all. If you must have a disk with the current release, we'll charge you \$2.75 for each disk (you need purchase only one diskette regardless of the number of registrations you purchase). Note that purchase of a diskette is not required. There is no difference between the software we send you and the shareware version. Serialization allows you to remove the shareware code (sawbuck.dll). However if you do request a diskette and a newer version/release (may be a pre-release copy) is available, we will send you the newest version.

There is no printed manual. All documentation is provided in the form of an on-line help file (such as the one you are reading). In addition to keeping the costs low, this also means you always get up-to-date information. There are no read.me files containing last-minute changes.

### Updates

All SawBuck software sports a version number in the form 99.99 (you'll find it in the about box). The number to left of the decimal indicates the version number. Version numbers change when the software undergoes major revisions or enhancements. The numbers to the right of the decimal indicates the release number. Release numbers change when the version number changes or when minor enhancements are added. Bug fixes also carry new release numbers.

When you register SawBuck software, your registration is valid for any release of the version purchased. There will be a charge for version upgrades. I have not yet set that charge (nothing has yet gone on to a second version), but it will be less than full price. I should also point out that SawBuck will not revise version numbers when adding minor features.

This page intentionally left blank.

**Qmenu 1.12 Registration Form**

Complete this form and mail or fax (01) 913 539-0348 with payment to:

SawBuck Systems  
P.O. Box 1172  
Manhattan, KS 66502-0012  
U.S.A.

Name: \_\_\_\_\_ (for registration)

Ship To: \_\_\_\_\_

Address: \_\_\_\_\_

Address: \_\_\_\_\_

Address: \_\_\_\_\_

Phone (optional) \_\_\_\_\_

E-Mail(optional) \_\_\_\_\_

I am registering \_\_\_\_ copies of Qmenu at \$10 each  
\_\_\_\_\_

Send me \_\_\_\_ (5 ¼ or 3 ½) (LD or HD) diskettes at \$2.75 each \_\_\_\_\_  
(optional)

\_\_\_\_\_ overseas add \$1 per disk for air  
mail \_\_\_\_\_

Subtotal \_\_\_\_\_

KS residents please include 6.4% sales tax

\_\_\_\_\_  Check  Money Order  Visa  MC Total \_\_\_\_\_

CC# \_\_\_\_\_ exp date: \_\_\_\_/\_\_\_\_

Signature \_\_\_\_\_ (for credit card purchase only)

List below the names to be registered (if different from above) or additional comments:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



Click one of the Move buttons to move the highlighted menu line up or down (<ALT-U> and <ALT-D>, respectively) one line. This simply affects the order in which the menu lines are displayed. This may be useful since the drag-and-drop method of adding lines always appends them to the end of the menu. These functions are disabled when the menu list (on the left) has the focus.



Click the Save button with your mouse or press <ALT-S> to immediately save changes made to the currently selected menu lines. The save function affects only changes made to menu lines, menu changes are effective immediately. This function is provided mostly for your peace of mind. All unsaved changes are automatically saved whenever you take an action that would erase the current contents of the menu lines list box. This would include selecting another menu to edit and closing the dialog box.

Likewise, the Undo button <ALT-N> will undo those changes.

Windows can have many applications (called tasks) running at seemingly the same time. Most tasks have a window(s) that is used for input and output. While any task can write to its window at virtually any time, only *one* task can receive keyboard or mouse input at any given time. This task is the active window and it (or one of its child windows) is said to have the *input focus*. If you have set up your colors properly in Control Panel, the active windows title bar (and/or border) will have a different color than other windows.

## How to..

**Change Qmenu Options**

**Drag and drop new menu items**

**Read/edit the QMENU.INI file**

**Resize the edit buffer**

**Start Programs when Windows starts up**

**Use Qmenu in place of Program Manager**

**Use the Menu Editor**

## Menu Entry

This is the menu line dialog. This is very similar to creating a new program item in program manager, and the fields are pretty much the same. If you need more information click below on the item in question.

**Execute:** The name of the program to run. If the program file is located along your PATH, it need not have the full path name (as in the example). This may also be a document file that has a program associated with it in File Manager (or registration database). For instance, an entry of `c:\windows\readme.wri` would invoke Windows Write with that file. The Browse button to the right can be used to help locate a program or file.

You can also select any of the Qmenu special functions or a menu name by clicking on the small arrow to the right or pressing the cursor up or down key.

**Menu Name:** This is the text that will appear in your menu.

**Params:** Parameters to pass to the program. In the example, we pass two file names to EW (an editor) which causes it to load those files when it starts up. If this were DOS, it would be the same as typing `c:\ew\ew win.ini system.ini` at the DOS command prompt. Valid parameters are entirely the discretion of the program you are running.

**Initial directory:** The directory to log onto before running the program. This will be the invoked programs default directory. This is the same as typing `cd xxxxx` at the DOS prompt with the exception that it can also changed the logged drive.

**Window:** This determines the window that the program will be run in. 99% of the time it will be normal allowing the invoked program to determine its window size (this has no effect on DOS programs). Other options are [min](#), [max](#), and [icon](#). Note that programs are not required to follow this directive (Qmenu itself, will disregard it), although most Windows programs will abide.

Click the mouse on the Browse button or press <ALT-B> to locate a program or file using the Windows open file dialog.

## Licenses

Sawbuck provides 3 basic license types:

The personal license is available for individual users and small *sole proprietorship* businesses (up to five computers). This license entitles you to install and use the registered software on all the computers in your (or small business). You may also install and use the software on one computer that is used exclusively by yourself at your place of employment.

A corporate license is available for businesses that require installation on multiple computers. This license provides a single serial number for use on many machines. Call or write [SawBuck Systems](#) for quantity discounts.

Finally, a site license is available for very large corporations. This is similar to the corporate license except that we provide you with a special edition of Qmenu that does not require serialization. This simplifies distribution among the corporate users. Once again, call or write for pricing.

The close button <ALT-O> closes the dialog box. You will be prompted to save any unsaved menu line changes, at which time you may elect to save, discard, or return to the dialog. Pressing the <ESC> key will have the same effect.

Printer Setup: This functions invokes the common dialog for Printer Setup. This sets the **default** printer settings. Please note that many programs (such as Word, Write, etc) can change these settings as needed.



## Startup Menu

When Qmenu is used as the Windows shell, it will automatically run the programs listed in the menu called Startup. By default, Qmenu gives this name to Menu41, although any menu will do. You can change this via the [menu editor](#). One thing this allows you to do is to maintain multiple startup menus. For instance, you might name 3 menus Startup, Startup2, Startup3. To select a particular one, simply go into the menu editor and change the name of the desired group to Startup. Be sure to rename the original menu something else. Qmenu will only run the first menu named startup in the MenuNames section of [qmenu.ini](#). You can do this manually by editing the MenuNames section of qmenu.ini.



