

## TrayMenu Version 1.3.3

Copyright (c) 1997, 1998 by Wei Ke, Iglyph Design, all rights reserved

### Version 1.3.3

What's new

Frequently Asked Questions

Description

How to install

How to use

Home folder and definition object

Mouse operation

Make shortcuts

Try it

Configure it

Create desktop shortcut

Organize shortcuts

Specify icon for folder as definition object

Specify shortcut display order

Insert menu separators

Identify an item (definition object)

Use attribute specifier

Command line options - to pop up the home menu and any menu defined by a definition object

About "TrayMenu Run" dialog box

How to uninstall

Comments on version 1.3.3

Special thanks

Package

License

Warranty

Feedback

### **New in version 1.3.3 Build 0921 [21-SEP-1998]**

- Fix the problem that a menu won't disappear when you click elsewhere under Windows 98.
- "Hide Top" function added.

### **New in version 1.3.3 [30-AUG-1998]**

- Multiple column menu support.
- Option to enable single click to open a sub menu item via "Catch mouse click on sub menu" feature.
- Option to hide "Open (##)" item, to be used with the above "single click" feature.
- Option to show special folders in desktop shortcut's context menu.
- Window list virtual shortcut to show/hide top level windows.
- Identified items to activate any pop up menu defined in home folder from command line.
- Allows specially named shortcut to specify icon for subfolders in home folder, it's now possible to use an icon stored in an exe or dll file.
- Problem with empty "Send To" item in a file's context menu is solved.
- Clicking both mouse buttons will dismiss popped up menus more reliably.
- Fine tuned code to eliminate some compatibility problems, including problems with ZipMagic98 and Windows98. The program is more stable and, as a side effect, more compact now.

### **New in version 1.3.2 [22-DEC-1997]**

- Starts a file in its directory.
- Allows Shift-Double Click/Click to open a shortcut to folder in the tray.
- Allows multiple instances by renaming the TrayMenu executable, for example, copy the file "TrayMenu.exe" to "TrayMenu2.exe", then you can run both "TrayMenu.exe" and

"TrayMenu2.exe" as two different applications. Note that if you place the renamed copy of the executable file to another location, you need to copy the help file as well WITHOUT renaming it.

#### **New in version 1.3.1** [12-NOV-1997]

- Added the option to repair damaged registry entries which can cause TrayMenu fail to recognize shortcut files. This fix is finally here.
- Fixed the bug of unable to recognize certain new shortcut file types such as "\*.URL".
- Added "TrayMenu Run" dialog box, you can specify additional command line parameters for the application being launched.
- Added "Apply" button to the properties dialog box.

#### **New in version 1.2.3** [14-OCT-1997]

- Fixed the bug of unable to execute "Copy/Cut" commands from an object's context menu which is called by TrayMenu.

#### **New in version 1.2.2** [12-OCT-1997]

- Fixed the bug of unable to read shortcut files on certain systems.
- Allows directory sorting method and file extension limits to be pre-specified (but not dynamically) in the attribute specifier.
- Option to hide shortcut overlay of shortcut to non-file system objects such as printers, disk drives, etc.
- Command line options allow messages to be passed to the running copy of TrayMenu to call menu commands and to activate TrayMenu's home menu from command line. You can store these command lines to shortcut files and use third party hot key tools to bind hot keys to them.

#### **New in version 1.1.1** [24-JUN-1997]

- Allows double-click to open folder menu items.
- Allows "Home Folder" menu item to be expanded into submenus.
- Option to allow menu separators to be displayed in the tray as blank icons.
- Opens shortcut using its specified show command.

#### **New in version 1.0.9** [06-JUN-1997]

- Option to hide Windows desktop.
- Added separator attribute, allows you to add separators in popup menu.
- Fixed the bug in loading icon specified in icon file.
- Option to remove "TrayMenu" from tool tips.

#### **New in version 1.0.7** [26-MAY-1997]

- Allows open of shortcut to desktop.
- Added "Windows Explorer" and "Display Properties" items to context menu of shortcut to desktop.
- Option to display default icon for shortcut to desktop, you don't have to specify a proper icon for it manually.
- Added "Browse..." button to "Enter custom home folder" edit box.

#### **New in version 1.0.6** [19-MAY-1997]

- Handles 32-bit menu handles correctly.
- Option to allow single click to open shortcuts in the tray.
- Option to dither gradient bar under 16-bit hicolor mode to avoid color bands.
- Option to hide shortcut overlay icon.
- Supports Ctrl-selecting and Ctrl-Shift-selecting folder item to explore and root-explore the folder.

### **Description**

This is a very powerful Windows9x toy. As many other taskbar tray utilities, it uses tray notification area as its workspace. TrayMenu simply puts icons of user specified shortcuts into the taskbar tray and lets the user launch those shortcuts or open their context menus. As the name implies, menus play a quite major role in TrayMenu. Basically, the real function that TrayMenu performs is to expand folder shortcuts, either

file system folders or virtual shell folders, into cascaded menus. Here are some of the features of TrayMenu,

- Explore your computer: the real fun of using this utility is that you can explore virtually everything (the only limitation is one at a time) from the menu activated by TrayMenu. If you specify a shortcut to a folder (including My Computer and even the desktop itself), and click its icon in the tray, you get a popup cascaded menu with all subfolders and files listed as menu items, subfolders can be accessed via submenus. So you can navigate the entire folder through this menu. You will see that TrayMenu is somewhat a menu explorer.
- Organize shortcuts: instead of consuming too much tray space, you can also organize your shortcuts by putting them into subfolders of TrayMenu home folder, in this case, you get subfolder icons instead of shortcut icons, clicking subfolder icon brings up a menu that lists all those shortcuts within the subfolder. You may use TrayMenu this way as an alternative "Start" button.
- Command line support: almost every function and every shortcut can be called out from the command line, you can use this feature to easily bind TrayMenu features with third party mouse or keyboard utilities to access your system even more efficiently, such as to pop up a menu that contains all the control panel applications via a hot key.
- Enhanced mouse operations: a user can use the mouse more effectively when navigate throughout a menu session, TrayMenu can detect various mouse event combinations on a menu item to perform different tasks. Such as Ctrl-click, Ctrl+Shift-click, left click, right click, left+right click and double click.
- Don't count it by size: although TrayMenu is very small, it's very powerful, you have to use it to believe it.

### **How to install**

I don't provide an installation utility because such a utility may be even bigger than the program to be installed. In fact, installation is very simple for TrayMenu, just 2 steps:

- 1) Create a folder anywhere in your hard disk (don't try to put it in the root, doing so will cause errors.), this folder is referred to as program folder in the context.
- 2) Copy "TrayMenu.exe", "TrayMenu.hlp" and the sample shortcut files "Desktop.desktop.lnk" (in Windows Explorer the extension ".lnk" is not shown) and "Window List.windowlist.lnk" to program folder. The sample files are not necessary to run TrayMenu, they are provided to give you some examples. If you are to keep these samples (in most cases you will), you should move them later to TrayMenu's home folder when you establish one.

Please ensure that no other files are copied, sometimes when you get TrayMenu not from its original package, e.g. from someone's running copy, there may be an ".ini" file. You should delete this file or, at least, disable the read-only attribute (typically when the file is copied from a CDROM), otherwise you will be unable to change settings of TrayMenu.

### **How to use**

To make a good use of TrayMenu is not that easy, you can use it in a very simple manner, but there are many powerful features under the surface. I have not enough time (to keep it free) and not enough space (to keep it small) to write a good user interface. But once you understand its concept, you'll find that it's very simple and flexible to customize TrayMenu and, in fact, TrayMenu totally relies on your customization. When you have configured it to your preference, you seldom need to alter the settings and just let it run in its peaceful way and use it all the time.

First you need to start the program "TrayMenu.exe" after you have installed it. Better put a shortcut to it in the startup folder so that it will start automatically when you start Windows. After you have started it, you'll see the TrayMenu icon in the taskbar tray if it's the first time you use TrayMenu, since you haven't added anything into its home folder. If you have copied the sample files, you will instead see two icons that represent the sample shortcuts.

**Home folder and definition object.** The folder where you store all the shortcuts for being used by TrayMenu as settings is called home folder. Initially, TrayMenu's home folder is the same as TrayMenu's program folder, you are recommended to create a dedicated home folder to save your settings. Home

folder can be set using the property sheet and be located using the main menu. Conceptually, shortcuts and subfolders containing shortcuts within home folder are called definition objects, TrayMenu uses these objects as the starting points of your navigation of the computer. TrayMenu treats shortcuts and other shell objects outside home folder as normal objects.

**Mouse operation.** There are some mouse tricks when playing TrayMenu, all are related to how to select a menu item.

- When "Catch left click on sub menu" option is checked, you can single click on a sub menu item to select it (execute command) instead of open the sub menu, otherwise, you should double click on it to select.
- When "Catch right click on sub menu" option is checked, you can right click on a sub menu item to open its context menu, otherwise, you cannot open a sub menu item's context menu.
- For a normal menu item that represents a shell object (such as a file), you can right click on it to open its context menu. Some context menus have special functions, such as the context menu of desktop shortcut and window list shortcut.
- When selecting an item, you can hold down the Ctrl key or the Ctrl and Shift keys, these are called Ctrl-select and Ctrl+Shift-select. They are often used to perform alternative actions on the item.
- Sometimes, you may get many cascaded menus from TrayMenu (for file navigation) that could cover the most part of the screen, in this you may feel hard to dismiss the menus. TrayMenu allows you to click both mouse buttons on the menus to dismiss them. This feature is very helpful under Windows98, since clicking outside a menu may not dismiss it.

**Make shortcuts.** From main menu, select "Home Folder" to open TrayMenu's home folder, you can create some shortcuts in this folder. You then see the TrayMenu icon disappears, instead there are icons of those shortcuts you have just created in the tray. If TrayMenu doesn't refresh the tray icon automatically, you can do it manually by selecting "Refresh" in the main menu. For better understanding, you should create at least 3 shortcuts, one to a folder, one to an executable program and one to a document file.

**Try it.** Now you can use TrayMenu to do something.

- Right clicking any shortcut icon in the tray brings up the context menu of the shortcut with "TrayMenu" submenu at the bottom. This submenu is called TrayMenu's main menu.
- Double clicking the executable shortcut or the document shortcut opens the shortcut. There is also an "Use single click to open shortcut" option to let you use single clicking instead. If you want to open the folder shortcut this way, you should hold down the Shift key while clicking the item.
- Left clicking the folder shortcut pops up a menu with all the objects in the folder listed, subfolder objects appear as submenu items, you can use the menu to navigate the entire folder. The item that says "Open (##)" on top of the menu stands for the folder, normally you can't select a submenu item unless you enable the "Catch mouse click on submenu" option, this item makes you be able to access the folder itself, also it is the default item of the submenu, double clicking on the submenu selects it. The number stands for the number of objects contained in the folder and the icon "N" - by name ("S" - by Size, "T" - by Type, "D" - by Date) shows the key used to sort items, the little arrow denotes ascending/descending. Sometimes you see an item that says "More (##)" at the bottom of a submenu if there are too many objects in the folder to be listed. You can open an object in the menu by left clicking it, or you can activate an object's context menu by right clicking it, if you want to close these menus, you can click anywhere within the menu using both buttons.
- When you try to open a folder, you have options to open it in various ways. Simply selecting the item opens the folder in normal folder view; selecting while holding down "Ctrl" key opens it in explorer view ("explore"); selecting while holding down "Ctrl+Shift" keys explores the folder using it as the root ("rootexplore" AKA "explore from here", this option is not available unless you've installed the MS PowerToy).
- When the Ctrl/Ctrl+Shift selecting trick applies to a file object, the "TrayMenu Run" dialog box pops up with the file name of this object filled into the "Edit command line" control. You can then specify additional parameters for this file object and launch it. Ctrl-selecting causes quoted long file name to be used, and Ctrl+Shift-selecting causes unquoted short file name (8.3) to be used.

**Configure it.** From main menu, you can select "Properties" to open the property sheet. Here you can specify your custom home folder by checking "Use custom home folder" checkbox and filling in the path of your custom home folder, it can be an absolute path or a relative path to the program folder. There are also some options available in the property sheet, for the description of each option, use the context help feature of the dialog box.

**Create desktop shortcut.** Normally you can't create a shortcut to desktop. TrayMenu lets you to specify this shortcut using a special name. You first create a shortcut to whatever anything, then you can change the name of this shortcut to `"*.desktop"` (e.g. `"shortcut to desktop.desktop"`), now this shortcut means *shortcut to desktop* to TrayMenu. By default TrayMenu displays the desktop icon for such a shortcut, this icon looks the same as the icon for desktop in Windows Explorer, if you want to display the icon specified in the shortcut, you need to uncheck "Use default icon for shortcut to desktop" option in the property sheet - "Special" tab. If you like, you can create multiple desktop shortcuts in home folder and/or its subfolders.

**Create window list shortcut.** A window list shortcut does not point to any shell folder, instead, it lets you popup a menu that lists all current top level windows. The way to define a window list shortcut is quite similar to desktop shortcut, conceptually, they are called virtual shortcut. The extension for window list virtual shortcut is `"windowlist"`. You first create a shortcut to whatever anything, then rename it to `"*.windowlist"` (for example `"Window List.windowlist"`). When you select a window list shortcut, it pops up a menu, each item in the menu shows the name and icon of a top level window. Selecting this item will show/hide the top level window and all of its children windows, Ctrl-selecting brings the top level window to the front, the operations of these two functions can be swapped by an option in the property sheet - "Special" tab, and Ctrl+Shift-selecting always minimize the top level window. The icon setting of window list shortcut can also be changed by unchecking "Use default icon for shortcut to window list" option in the property sheet.

**Organize shortcuts.** You can create subfolders in TrayMenu's home folder, and put shortcuts in the subfolders. These folders are also definition objects. Such a folder defines a menu that lists all the definition objects stored in the folder including nested subfolders. With this feature, you are able to organize your shortcuts in cascaded menus, the way you like.

**Specify icon for folder as definition object.** If you use a subfolder to organize your shortcuts, the default icon for the folder is a folder icon, but you can change this icon by putting an icon file (with file extension `".ico"`) in the folder, or, even more conveniently, you can put a shortcut file with `".ico"` file extension (visible extension, in front of the hidden shortcut extension e.g. `".lnk"`) in the folder, and the folder icon will be changed to the icon for this shortcut. You don't have to extract an icon out of an exe or dll file to use it for the folder.

**Specify shortcut display order.** You can arrange the definition object's display order by specifying display priority in the shortcut name, this feature is called attribute specifier, described below.

**Insert menu separators.** You can also insert menu separators above and/or below a definition object via attribute specifier, described below.

**Identify an item (definition object).** An identified item lets you pop up the menu defined by the item from command line using "-homemenu" option. An item's identifier is defined via attribute specifier, described below.

**Use attribute specifier.** Attribute specifier is a special file name extension to the definition object name. This extension is not displayed by TrayMenu, it is used to define various attributes of the definition object. Attribute specifier has the form: `" (#1, #2, #3, #4, #5, #6, #7) "` where `#x` denotes a decimal number.

#1: Primary display order, in range of -2147483648 to 2147483647, items with smaller value are displayed first, default value: 10000.

#2: Secondary display order, in range of -2147483648 to 2147483647, item with smaller value displayed first, default value: 10000.

The first two numbers specify the shortcut's display priority, one is primary and the other secondary. Since shortcuts are of different categories according to their targets, normally shortcuts to folders are always displayed before shortcuts to files. If you specify an explicit primary display order for a shortcut, the order is compared before the category, so using primary order specifier, you can display shortcuts to files before shortcuts to folders. The secondary order is compared after the category, so using secondary order specifier, you can arrange shortcuts of the same category.

**#3: Icon size in submenu**, in range of 0 to 1, default value: global setting. This number specifies whether to use large icons in the item's submenu if it contains one, if this number is 1, large icons are used, if this number is 0, small icons are used, this specifier overrides the global icon size settings found in the property sheet - "View" tab.

**#4: Menu separator settings**, in range of 0 to 3, default value: 0. This number tells TrayMenu whether to insert separators above and below this menu item.

- 0: no separator
- 1: separator above
- 2: separator below
- 3: separators surrounded

Only one separator can be displayed between two menu items, additional specified separators are ignored.

**#5: Sorting method setting**, in range of 0 to 3 and range of 8 to 11, default value: 0. This attribute applies to a shortcut to folder only. The number specifies which key to be used to sort its submenu items.

- 0: by Name
- 1: by Size
- 2: by Type
- 3: by Date

Adding 8 to these values forces TrayMenu to sort descending otherwise it sorts items ascending (e.g. 11 means to sort descending by date).

Please note that "Type" key means the type name associates with the file extension, not the file extension string itself, for example, "file.exe" is commonly registered as "Application" whose string value is smaller than "Application Extension" type name registered for "file.dll"; this behavior is consistent with the Windows Explorer.

Only file system objects (real files and directories) can be sorted by keys other than name, other shell objects such as control panel applets, printers are always sorted by their names.

**#6: File extension limits index**, default value: 0. When this number is greater than 0, it specifies starting from which dot (".") in the object name (based 1) in front of the attribute specifier is the file extension limits string, this string is truncated from the display name of this object and limits the submenu items to those files with the file extensions listed in the string. For example, "object.name.exe.dll.sys.(, , , , , 2)" specifies a definition object named "object.name" and a file extension limits string ".exe.dll.sys" which limits the submenu items to files with extensions ".exe", ".dll" and ".sys". A single character "%" in such an extension denotes empty extension name, while two consecutive "%" denotes a single "%", thus ".%.t%%t.%%%%%%%%" specifies three extension names: ". ", ".t%", ".%%". This attribute applies to folder shortcuts only.

**#7: Item identifier**, default value: 0. When this number is greater than 0, it assigns an identifier to the item, this identifier is referred to by the command line to locate the item in order to pop up the specified menu. For example, "shortcut to folder A.(, , , , , , 100)" assigns 100 as the identifier to item "shortcut to folder A", then you can use the command line "traymenu.exe - homemenu.100" to pop up the menu that lists all the files contained in "folder A". An identifier can be in range of 1 to 65535, that is a 16-bit non-zero value. Items with identifier are called identified items, they are summarized under the "Identified Items" item in the main menu.

You can omit any or all numbers in the attribute specifier extension. Examples: "shortcut1.(1,1,0,1)"; "shortcut2.(1,1,1).()"; "shortcut3.(, , , 3,10)"; "shortcut4.(, 20000)".

Please note that a virtual shortcut extension such as shortcut to desktop or shortcut to window list file extension is parsed and truncated before attribute specifier, so ".desktop" extension should be following the attribute specifier if you want to specify attributes for a desktop shortcut.

*This paragraph is for advanced users only.* As you would notice it, some definition objects are shortcut files, they have file extensions like ".lnk" or ".pif" or something else, although both Explorer and TrayMenu never display them. In this context, object name means its display name, it's the name you can alter by selecting the item and clicking on its name field. For example if a shortcut file's display name is "shortcut.ext.(,,,,,1).desktop", its actual file name may be "shortcut.ext.(,,,,,1).desktop.lnk".

### **Command line options - to pop up the home menu and any menu defined by a definition object**

Home menu contains items listed in the tray by TrayMenu. These items are in the top level of the home folder. Home menu is another way to access the defined objects. Once TrayMenu is running, you can start it again with command line options to activate the home menu at the current cursor position. The newly started TrayMenu can detect its previous instance and sent it an "activate the home menu" message and then simply quit. If you have assigned identifier to a definition object using attribute specifier, you can also pop up the menu defined by the definition object using this method, the syntax of command line is describe below.

There are also other command line options which can pass menu command messages to call menu commands from the command line, these menu commands are listed in the main menu, the context menu of desktop shortcut and window list shortcut.

The command line options are listed below:

-homemenu.small	Pop up the home menu with small icon.
-homemenu.large	Pop up the home menu with large icon.
-homemenu.###	Pop up the menu defined by the definition object with identifier ###, where ### denotes a decimal number.
-command.file_properties	Display TrayMenu's property sheet.
-command.file_homefolder	Open the home folder.
-command.file_refresh	Force TrayMenu to reload the home folder contents.
-command.file_hide	Show/hide TrayMenu in/from the tray.
-command.file_help	Open the TrayMenu help file.
-command.file_exit	Exit TrayMenu.
-command.desktop_explorer	Launch Windows Explorer.
-command.desktop_hide	Show/hide the shell desktop view (may not work on certain systems).
-command.desktop_run	Open "TrayMenu Run" dialog box.
-command.windowlist_closeall	Close all top level windows.
-command.windowlist_showall	Show all top level windows that have been hidden by TrayMenu.
-command.windowlist_hideall	Hide all top level windows.
-command.windowlist_hidetop	Hide the current top window, note: this function may not work on your system, it's OS version dependent.

For example, when TrayMenu is running, type "<TrayMenu path>.exe /homemenu.large" in the Run dialog box. You can also store this command line in a shortcut file and bind a hot key to the shortcut file.

### **About "TrayMenu Run" dialog box**

The "TrayMenu Run" dialog box is the place where you can specify additional parameters for the file being launched. When you Ctrl-select (or Ctrl+Shift-select) a file object from TrayMenu, this dialog box pops up with the file name (short form is used instead if Ctrl+Shift selecting) filled in the command line control, you can then modify this command line. This dialog box is a full functioning command line launcher in general, it can be opened from desktop shortcut's context menu or by the command line option "-command.desktop\_run". You can use "Browse..." and "Browse for short..." buttons to get full file names either in long form or short form, since some old DOS programs don't understand long file names, you must use "Browse for short..." in these cases. You can also drag'n'drop files on to this dialog box to retrieve file names, normally you get long file names unless you drop the files on to the "Browse for short..." button which lets you retrieve the short form of the file names. "TrayMenu Run" also supports multiple file selecting and dropping, passing alternative show command and command line history.

## How to uninstall

Uninstalling TrayMenu is very simple too, since it doesn't put anything automatically outside the TrayMenu program folder nor in the registry. Simply delete the TrayMenu program folder after you have exited TrayMenu.

## Comments on version 1.3.3

When pressing both mouse buttons to cancel a menu, you need only ensure that there is a period when both mouse buttons are down together. You needn't release them up to closely, in fact, doing so will make Windows confused with the button status and produce unexpected result (Windows will be unable to detect that the right button is up, in this case you can press right button again to reset the status).

There are also some options which can't be accessed via TrayMenu UI, these options are not recommended by the author. However I list them here for reference:

- [options] ScrollMenu=0 can be set to [0]1, this setting is for Windows98 users to use scrollable menu.
- [options] DefaultOrder=10000 can be set to any decimal number in range of -2147483648 to 2147483647, this setting specifies the default display order of a definition object.
- [options] SetWindowTopOnShow=1 can be set to [0]1, when set to 0, the z order of a window will not change if it is shown from hidden state, otherwise it will be brought to front.
- [options] HideHideTop=1 can be set to [0]1, when set to 0, there's is an item "Hide Top" in the window list shortcut's context menu that can hide the current top window. This feature can always be accessed by command line option "-command.windowlist\_hidetop".

These settings can be entered and altered in the "TrayMenu.ini" file. And for user specific settings, you can alter them in section [options.username] instead of [options], which is for global settings.

The most important fix in version 1.3.2 is the option to repair relevant registry entries. Normally, TrayMenu doesn't bother the registry, but TrayMenu depends on Windows to report it whether a file is a shortcut file. These system functions rely on certain registry entries to work. For some reasons (I don't know what they were) these registry entries were removed on some systems, and for this reason TrayMenu fails to recognize shortcut files on these systems. This problem has been reported since the first release of TrayMenu. When you open the properties dialog box, if you see that the "Repair relevant registry entries" check box is enabled, then it means TrayMenu has found those entries are missing, if you do encounter the problem that no shortcut icon is shown in the tray even if you have put some shortcut files in the home folder, it's highly recommended you try to repair the registry by checking this checkbox and select "OK" or "Apply". TrayMenu simply merges the following entries into the registry to do the repair.

```
[HKEY_CLASSES_ROOT\lnkfile]
"IsShortcut"=""
[HKEY_CLASSES_ROOT\piffile]
"IsShortcut"=""
```

When calling a TrayMenu command line to pass messages to the running copy of TrayMenu, another copy of TrayMenu will start and send some messages and then quit. This copy does not create any main window but just runs for a very short period, thus the Windows shell cannot track it. So if the command line is called by pressing a shortcut file's hot key (Windows' built-in function) and you hold down the hot key for a relatively long period, many hot key repeats will be generated and unexpected numbers of the TrayMenu command line will be called. Please use a better hot key tool, or release the hot key immediately.

Users of MoreAct Context Menu Extension (another freeware utility written by the same author of TrayMenu) need to download the new version (version 0.5.6 dated after 13-MAY-1997).

Since you can use TrayMenu to explore your entire computer, theoretically it's possible that TrayMenu is



running out of menu item identifiers (65536 maximum items per menu including its submenus). But in practice, since TrayMenu allocates menu items on demand (menu items in a submenu are allocated only when you activate the submenu) it's almost impossible for a user to view that many items in a menu session. If you exit from a menu session, all items are freed and resources can be reused next time.

If you create a desktop shortcut, context menu of this shortcut is not available, instead some commands related to the desktop are displayed. This is because the shortcut to desktop is a virtual shortcut that only lets you access items on the desktop.

The "Hide shell desktop view" option may not work on some systems. This is because TrayMenu uses a non-standard trick to search the desktop view, if it can't find the view by using this trick, this option is disabled.

### **Special thanks**

A special thanks goes to the author who made the "Tray Shortcuts" freeware - Clovis Lofton. He gave me the idea of using shortcuts to specify file paths other than writing them in a script file or the system registry. For end users managing shortcuts maybe easier.

And Ahto Tanner, the author of "Multimedia Xplorer", introduced me the ideas of Ctrl-selecting and default desktop icon. Travis Bauer and Andrew suggested me to add single click and hide shortcut overlay options. Wolfgang Krietsch helped me to correct menu handle errors and suggested me to add the option of insertion of menu separators. Jon Aske helped to find the error in loading icon from icon file. Jan Goldreich gave me the idea of using blank icons as separators. Frédéric Jean suggested me to use a stand alone program to pass messages to the running TrayMenu. Stavros Goulionis suggested me the "WindowList\_HideTop" feature. And the many good ideas (for instance: hide shortcut overlay of non-file system object) from Paul Richards made me think of the design more carefully.

And Crawford Dales helped me to find out the cause of a "classic" TrayMenu problem - unable to recognize shortcut files on some Windows systems. He also suggested me some good ideas to solve it.

Thank all who gave me bug reports, suggestions, messages and feedback.

### **Package**

TrayMenu 1.3.3 is a software package with four files in it, they are the program file "TrayMenu.exe", the help file "TrayMenu.hlp" and the sample files "Desktop.desktop.lnk", "WindowList.windowlist.lnk".

### **License**

You may use TrayMenu freely on your computer system. You may distribute TrayMenu freely on online services so long as the files in this package are kept together and unmodified. You may not sell copies of TrayMenu. You may not decompile, reverse engineer, or modify TrayMenu or its resources. You may not modify the files "traymenu.exe" and "traymenu.hlp".

### **Warranty**

This program (TrayMenu) is provided "as is", use of this program is at your own risk. You agree by using this program to hold harmless the developer, Wei Ke, against damages arising out of the use of this program.

### **Feedback**

Any suggestions or bug reports are welcome, please contact me:

Wei Ke

Internet email: [kew@163.net](mailto:kew@163.net)

**As a freeware author, I expect feedback, please support me by simply sending me bug reports, suggestions, ideas and messages.**

**Main menu** contains some TrayMenu commands related to the program execution and maintenance, it's usually called "File" menu in other applications. To pop up main menu, you can right click on any tray icon displayed by TrayMenu, then open the "TrayMenu" sub menu at the bottom.

**Home folder** is the folder where you store all the shortcuts for being used by TrayMenu as settings. Initially, TrayMenu's home folder is the same as TrayMenu's program folder, it can be set using the property sheet and be located using the main menu.

**Use gradient menu bar**

Checking this option causes TrayMenu to paint menu selection bar using gradient color, the painting starts with standard selection color and fades into menu background color. You should uncheck this if you want standard menu selection bar.

**Use large file icon**

Checking this option causes TrayMenu to display large file icon with menu item in menus activated by shortcuts to folders, otherwise it displays small file icon.

**Use large shortcut icon**

Checking this option causes TrayMenu to display large shortcut icon with menu item in menus activated by subfolders in the TrayMenu home folder, otherwise it displays small shortcut icon.

**Use button-like icon effect**

Checking this option causes TrayMenu to display icon on a button-like background.



**Use custom home folder**

Checking this option causes TrayMenu to search shortcuts in the user specified home folder, otherwise TrayMenu searches shortcuts in its program folder.

You specify your custom home folder here, the path name of home folder can be either a full path or a relative path to the TrayMenu program folder. You should also check the "Use custom home folder" checkbox to use it.

**Show shortcut overlay**

Checking this option causes TrayMenu to display the "Shell Shortcut Overlay Icon" (typically a small arrow) on top of each shortcut item icon, this option applies only to the items within TrayMenu home folder (definition objects).

**Show shortcut overlay of non-file system object**

Unchecking this option causes TrayMenu to hide the "Shell Shortcut Overlay Icon" away from icon of shortcut item whose target is not a file system object, such as printers, disk drives, control panel applets, etc. This option does not apply to definition objects.

**Dither gradient bar under 16-bit hicolor mode**

Checking this option causes TrayMenu to dither gradient menu bar when using 16-bit hicolor display mode, this produces better gradient effect and avoids color bands. Don't use this option if it slows down the drawing speed or produces unexpected result. Under other display modes, this option has no effect.

**Use single click to open shortcut**

Checking this option allows you to open shortcut to file within the tray by a single click of left mouse button, otherwise you should double-click to open shortcut to file within the tray.

**Use default icon for shortcut to desktop**

Checking this option causes TrayMenu to display the default icon for shortcut to desktop, this icon is the icon for desktop in Windows Explorer. If you want to display the icon specified in the shortcut, you should uncheck this option.

**Hide shell desktop view**

Checking this option causes TrayMenu to hide the Windows desktop view, you should use this option with caution. Since TrayMenu uses a non-standard trick to search the desktop view, this option may not work on your system.



**Hide TrayMenu from tray**

Checking this option causes TrayMenu to hide itself away from the tray. You may still be able to access TrayMenu items from the home menu. See the help file for more information on command line options and home menu.

**Show "TrayMenu" in tool tip**

Checking this option causes TrayMenu to display "TrayMenu" following the item name in its tool tip, this is to provide a way to determine whether the tray icon is displayed by TrayMenu. You can uncheck this to remove "TrayMenu" from the tool tip.

**Show separators in tray**

Checking this option causes TrayMenu to turn menu separator settings into blank tray icons if these settings belong to items displayed in the tray.

**Exit TrayMenu**

Checking this option causes TrayMenu to exit when you select OK or Apply button to save the changes.

**Repair relevant registry entries**

Checking this option causes TrayMenu to repair damaged registry entries which can cause TrayMenu fail to recognize shortcut files, grayed option means that your relevant registry entries are good. The action will take effect when you select OK or Apply button to save the changes.

Browse for custom home folder.

You specify the maximum menu width here, the width is in percent of the screen width. When a menu exceeds this width, it won't add more columns, instead, it will add a "More" item at the end.

You specify the maximum number of menu columns here, a number between 1 and 128 is effective, if it's out of range, the nearest effective one is used.  
you can specify 1 to disable multiple column menu feature.



**Catch right click on sub menu**

Checking this option causes TrayMenu to catch the click on a sub menu item in order to activate context menu of the menu item instead of open the sub menu. Normally the sub menu will open when you move the mouse cursor onto it, so this option allows you to reuse the click to improve productivity. Some non-standard tricks are used to implement this, if you encounter problems, you should uncheck this to retain the standard menu system behavior. See also [Include "Open \(##\)" item in folder sub menu](#)

**Catch left click on sub menu**

Checking this option causes TrayMenu to catch the click on a sub menu item in order to launch the menu item instead of open the sub menu. Normally the sub menu will open when you move the mouse cursor onto it, so this option allows you to reuse the click to improve productivity. Some non-standard tricks are used to implement this, if you encounter problems, you should uncheck this to retain the standard menu system behavior.  
See also [Include "Open \(##\)" item in folder sub menu](#)

**Use Windows 95 menu alignment**

Checking this option causes TrayMenu to align pop up menus in the old way, this may cause some menus to be displayed out of the screen under Window 98. It is recommended that you uncheck this option.

**Enable shortcut with ".ico" extension to specify icon**

Checking this option causes TrayMenu to display the icon specified in a shortcut file with ".ico" extension for the shortcut's parent folder.

**Include hidden files**

Checking this option causes TrayMenu to display hidden files and folders in the menu.

**Align popup menu to tray window**

Checking this option causes TrayMenu to align menus popped up from tray to the border of tray window.

**Bold default menu item**

Checking this option causes TrayMenu to use boldface for default item in folder menu, typically the "Open" item. Standard menus are not affected.

**Include "Open (##)" item in folder sub menu**

Checking this option causes TrayMenu to show an "Open" item on top of each folder menu. When "Catch right click on sub menu" function is disabled, this option is necessary to activate the context menu of a folder item.



**Use default icon for shortcut to window list**

Checking this option causes TrayMenu to display the default icon for shortcut to window list. If you want to display the icon specified in the shortcut, you should uncheck this option.

**Set show/hide as default action on window items**

Checking this option causes TrayMenu to show/hide a window when you select the corresponding item in the window list, otherwise, TrayMenu brings the window to front. You can hold down Ctrl key while selecting an item to trigger the alternative.

**Include menu for window items**

Checking this option causes TrayMenu to attach the context menu of a window item to it as a sub menu. You can always access this context menu by right clicking an item in window list.

**Group hidden windows in window list**

Checking this option causes TrayMenu to display all hidden windows in the lower part of the window list menu, otherwise, all windows are displayed in the order of their window titles.

**Hide "Close All" from window list context menu**

Checking this option causes TrayMenu to hide "Close All" item from window list context menu to prevent the user from accidentally closing all the windows.

This section specifies which special folders to be displayed in the desktop shortcut's context menu.

**Control Panel**

Virtual folder containing icons for the control panel applications.

**Desktop Directory**

File system directory used to physically store file objects on the desktop (not to be confused with the desktop folder itself).



**My Computer**

Virtual folder containing everything on the local computer: storage devices, printers, and Control Panel. The folder may also contain mapped network drives.

**NetHood**

File system directory containing objects that appear in the network neighborhood.

**Network Neighborhood**

Virtual folder representing the top level of the network hierarchy.

**My Documents**

File system directory that serves as a common repository for documents.

**Printers**

Virtual folder containing installed printers.

**Programs**

File system directory that contains the user's program groups (which are also file system directories).

**Recent**

File system directory that contains the user's most recently used documents.

**Send To**

File system directory that contains Send To menu items.



**Start Menu**

File system directory containing Start menu items.

**Startup**

File system directory that corresponds to the user's Startup program group.

**Recycle Bin**

File system directory containing file objects in the user's recycle bin.

You specify your command line to execute here. This command line can include command and parameters up to 16k characters long. You can specify filenames by dropping files onto this dialog box or using "Browse..." and "Browse for short..." buttons.

The URL of TrayMenu's home page.

The email address for TrayMenu feedback.

Memory usage information.

Click this to start TrayMenu's home page.



[Click this to send feedback via email.](#)

You specify how the application window is to be shown here, if any. Although your suggestion is sent to the application being launched, the application doesn't have to use it.

Click this to browse through folders to find the files you want. This function will return quoted long file names. Use "Browse for short..." for short file names.

Click this to browse through folders to find the files you want. This function will return unquoted short file names (8.3). Use "Browse..." for long file names. Since you can drop files onto this dialog box, if you drop onto this button, you will get short file names, otherwise you will get normal long file names.

**Keep open**

Checking this option leaves "TrayMenu Run" open when you select "OK" to launch a file, this allows you to continue to use the dialog box.

TrayMenu icon.

Product name and copyright information.

## TrayMenu Version 1.3.3 Frequently Asked Questions

Copyright (c) 1997, 1998 by Wei Ke, Iglyph Design, all rights reserved

- [What can I do with TrayMenu?](#)
- [Why doesn't TrayMenu recognize shortcut files in its home folder on my system?](#)
- [How can I specify the icon for a subfolder in the home folder?](#)
- [How can I get rid of those ugly tiny arrows upon my shortcut's icon?](#)
- [I don't like to see the tiny arrows on icons of shortcut to "Drive", "Printer", "Control Panel" etc. while keep the arrows on icons of shortcut to real files and folders. How to do this?](#)

### **What can I do with TrayMenu?**

- Your start menu maybe too crowded. You can use TrayMenu as additional start menus.
- You can store shortcut to folders in TrayMenu's home folder (or its sub folders) to use cascaded menus to navigate those folders. You can access virtually every object visible in the explorer using this method.
- You can specify additional command line parameters when launching an application using TrayMenu (Ctrl-selecting).
- You get miscellaneous functions that can make your desktop life easier.
- Every function introduced in TrayMenu is implemented as complete as possible, try them thoroughly.

### **Why doesn't TrayMenu recognize shortcut files in its home folder on my system?**

This is a classic problem which has been there since the beginning. Some registry entries used by the system to determine shortcut files were removed by certain unfriendly programs. These entries were installed by Windows, but TrayMenu can repair them. If you encounter this problem, try to open the TrayMenu properties dialog box (right click any TrayMenu icon, select TrayMenu/Properties), if you see the "Repair relevant registry entries" checkbox not being grayed out, then check it and select "OK" or "Apply" to repair the registry, and then you should have solved the problem. See [comments](#) for more.

### **How can I specify the icon for a subfolder in the home folder?**

Simply copy an icon file "\*.ico" into the subfolder to which you want to assign the icon. See [Organize shortcuts](#) for more.

### **How can I get rid of those ugly tiny arrows upon my shortcut's icon?**

Uncheck the "Show shortcut overlay" checkbox in the TrayMenu properties dialog box.

### **I don't like to see the tiny arrows on icons of shortcut to "Drive", "Printer", "Control Panel" etc. while keep the arrows on icons of shortcut to real files and folders. How to do this?**

Uncheck the "Show shortcut overlay of non-file system object" checkbox in the TrayMenu properties dialog box.



