

TrayMenu Version 1.2

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

Version 1.2

[What's new](#)

[Description](#)

[How to install](#)

[How to use](#)

[Home folder and definition object](#)

[See it](#)

[Make shortcuts](#)

[Try it](#)

[Configure it](#)

[Create desktop shortcut](#)

[Organize shortcuts](#)

[Specify shortcut display order](#)

[Insert menu separators](#)

[Use attribute specifier](#)

[Command line options - to pop up the home menu](#)

[How to uninstall](#)

[Comments on version 1.2](#)

[Special thanks](#)

[Package](#)

[License](#)

[Warranty](#)

[Feedback](#)

New in version 1.2.3

- 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

- 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

- 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

- 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

- 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

- 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 Windows95 toy. As many other taskbar tray utilities, it uses tray notification area as its workspace. TrayMenu simply puts icons of user specified shell shortcuts into the taskbar tray and lets the user launch those shortcuts or open their context menus. If you have ever used another tray utility called "Tray Shortcuts" (by Clovis Lofton), you should have the concept. So why TrayMenu? Here are the reasons:

- 1) 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), when you click its icon in the taskbar tray, you get a popup hierarchical menu with all subfolders and files of that folder as menu items, subfolders can be accessed via submenus. So you can navigate the entire folder through this menu. You will see this TrayMenu is somewhat a menu explorer.
- 2) Detect mouse buttons: selecting an item by left button opens the file or folder, by right button opens the context menu of the file or folder, and selecting by both buttons cancels the menu.
- 3) 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 includes all those shortcuts within the subfolder.
- 4) 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, this has never been tested.), this folder is referred to as program folder in the context.
- 2) Copy "TrayMenu.exe" and "TrayMenu.hlp" to program folder.

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 you'll still be able to enjoy it once you get into it. Actually when you have configured it to your preference, you seldom need to reconfigure it 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 install it. Better put a shortcut to it in the startup folder so that it will start automatically when you start Windows. After you have start 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.

Home folder and definition object. The folder where you store all the shortcuts for being used by TrayMenu is called home folder. Initially, TrayMenu home folder is the same as TrayMenu program folder.

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.

See it. Right click the TrayMenu icon, you get a popup menu with "TrayMenu" submenu item at the bottom, double-click "Home Folder" item from this submenu to open 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" form "TrayMenu" submenu.

Make shortcuts. 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.

- 1) Right clicking any shortcut icon in the tray brings up a context menu of that shortcut with "TrayMenu" submenu at the bottom.
- 2) Double clicking the executable shortcut or the document shortcut opens the shortcut. There is also the "Use single click to open shortcut" option to let you use single clicking instead.
- 3) Left clicking the folder shortcut pops up a menu with object names in the folder as menu items, subfolder objects appear as submenu items, you can use this menu to navigate the folder. You notice that there is an item with a name says "(####) by Name" on top of each menu and submenu, this item stands for the folder containing the items beneath it (since you can't select a submenu item, this item lets you access the folder itself, also this item is the default item of the submenu, so double clicking on the submenu selects this item), the number stands for the number of objects contained in the folder and "by Name" ("by Size", "by Type", "by Date") shows the key used to sort items, the little arrow on the left side stands for ascending/descending. Also perhaps you see an item with a name says "(####) More" at the bottom of some submenus if there are too many objects in the folder to be listed in a single menu. Now you can open any object in the menu by left clicking it, or you can activate any object's context menu by right clicking it, if you want to close these menus, you can click any non submenu item using both buttons.
- 4) When you try to open a folder object by selecting the "(####) in Folder" item, 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").

Configure it. Right clicking any icon created by TrayMenu in the tray, you can access the "TrayMenu" submenu from the popped up context menu. Select "Properties" to open the properties dialog box. Now you can specify your custom home folder by checking "Use custom home folder" checkbox and entering your custom home folder path, this path can be an absolute one or a relative path to the program folder. There are also some options available in this dialog box, 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 the 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 a default 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 TrayMenu properties dialog box. If you like, you can create multiple desktop shortcuts in home folder and/or its subfolder.

Organize shortcuts. You can create subfolders in TrayMenu home folder, and put shortcuts in subfolders. If you put an icon file in the subfolder, this icon will appear in the tray (or popup menu) to identify the subfolder.

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.

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 shortcut.

Attribute specifier has the form: ". (#1, #2, #3, #4, #5, #6) " where #x denotes a decimal number.

#1: Primary display order, in range of -2147483648 to 2147483647, item with smaller value 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 submenu of the shortcuts 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 properties dialog box.

#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". This attribute applies to a shortcut to folder only.

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 shortcut to desktop 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

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 root 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.

There are also another command line option which can pass menu command messages to call menu commands from the command line, these menu commands are listed in the "TrayMenu" submenu and the desktop shortcut's context menu.

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.
-command.file_properties	Display the TrayMenu properties 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_properties	Display the display properties sheet (may not work on certain systems).

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.

How to uninstall

Uninstalling TrayMenu is very simple too, since it doesn't put anything automatically outside the TrayMenu program folder or in the registry. Simply delete the TrayMenu program folder after you have exited TrayMenu.

Comments on version 1.2

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 keys 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 Microsoft's SendToX power toy are not able to access SendToX submenu from the context menus activated by TrayMenu, the cause is unknown at the moment. This issue is under studying.

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 13th of May '97).

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 hierarchy). 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 per menu active. If you close a main popup menu, all items are freed and resources can be reused for the next menu active.

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.

Shortcuts to Network Neighborhood are supported by design, but never tested. If you encounter problems please send email to the author.

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.

The "Display properties" menu item may not be available on some systems especially non-English versions of Windows. This is because TrayMenu uses the name "Display" to search the display control panel, if it can't find the control panel by using the name "Display", this menu item is not shown in the shortcut to desktop context menu from the tray.

Special thanks

Again, 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. 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.

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

Package

TrayMenu 1.2 is a software package with two files in it, they are "TrayMenu.exe" and "TrayMenu.hlp".

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 "trmn123.zip" file or its contents.

Warranty

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: zhkw@pub.zhuhai.gd.cn

As a freeware author, the only thing I can expect is feedback, please support me by simply sending me suggestions, ideas and messages.

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.

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 button to close this properties dialog box.

Browse for custom home folder.

TrayMenu icon.

Product name and copyright information.

