

History of RRKMenu 1.1  
History  
of  
New Menus for Windows 1.1  
© 1993 Roger René Kommer

**New Menus for Windows 1.1g/e/i (German, English, International)**

Changed: The file RRKSHELL.EXE is now unnecessary.

Changed: The Internal Commands (MNU\_...) are now all in English.

Changed: The files RRKMNDLL.DLL and BWCC.DLL must be in the Windows directory.

Changed: If you arrange all sticked Menus on the desktop (right mousebutton over a sticked Menu or F9 resp. F11), the Menus, which only shows the icon and the sticked Drop-Down-Menus aren't effect.

Changed: The option **Save Wins DT**, which appears in the Submenu Tasks if the Menu is the Win-Shell, is changed: The option can now reset with the Shift-key. The internal command MNU\_SD T is now public usable.

NEW: Also the Shortcuts for Submenus (Ctrl - char) are definable in the Editdialogue for Menuitems.

NEW: If you doubleclick the title bar of a sticked Menu, the marked menuitem will be executed. This also function, if the Menu is rolled up or if the menuitem is a (link to a) Submenu. If you press the **Shift**-key while selecting a Menuitem, a sticked Menu will rolled up after executing the Menuitems command.

NEW: Implemented global hotkeys to execute command lines (also internal commands). Hotkeys can be defined in the edit dialogue of a menuitem or in the dialogue **Hotkeys**. The Hotkeys only function, if the switch **Alt+Pause-Popup (Hotkeys)** is activ.

NEW: Reorganised the edit dialogue for Menuitems. If you edit Submenu - items, in the cathegorie Window-Size you can define Submenus as normal, only with icon, only with text, with no pin and locked.

NEW: Now you can also popup the Task-Menu with an hot-key (The **Ctrl-X** key is preselect). This option has only an disadvantage: Depending of the time pressing the **Ctrl**-Key in the Task-Menu only the visible Windows are showed or also the hidden.

FIXED: The Menu was confused by parameters of a PMG-Menuitem - Not any more.

NEW: If you press the **Ctrl**-key, while opening a Submenu or a Drive Directory, the current Submenu will popuped as the main-Menu. This option may improve the possibilities to walk through your directories. (There are some problems if WinWord 2.0 has the focus.)

NEW: With the internal command **MNU\_KB** you can send keys to the window, which has the focus. Unfortunately this doesn't function with DOS-Boxes. (It is only a first hack and not very comfortable.)

NEW: **Now you can stick every Windows Drop-Down Menu with a pin on the desktop!** Do this, select a Drop-Down Menu not with the left mousebutton, but with the right mousebutton.

NEW: The Menu can popuped with external program. The function is

```
int CallRRKMenu(char far *Submenu);
```

and can be found in the file RRKMNDLL.DLL. For more hints look in the help in

## History of RRKMenu 1.1

the chapters about the API of RRKMenu.

NEW: Instead of a filename to open or to execute you can declare a function of the Windows-API or another DLL with the internal command **MNU\_CALLDLL**. For more hints look in the help in the chapters about the API of RRKMenu.

NEW: RRKMenü becomes more and more an Windows-Manager! You can associate special Popup-Menus to Windows-Classes and improve the functionality and comfort of Windows. The way to assign Menus with special Window-Classes is described in the chapters about the API of RRKMenu.

I've done it with edit-controls, list-boxes, scrollbars and the DOS-Box. If you press the right mousebutton over one of this Window-Classes a Menu appears to manipulate it. For more hints look in ADDONS.WRI.

NEW: If you press the Alt-key while dropping the files, the menu will start for each file the application. This is useful, because most of the applications, which only can handle one document (f.e. Notepad or Paintbrush), also accept only one file as argument.

Otherwise the files are send as one package to the application.

NEW: If you quit the Menu with **Menu Setup / End** while pressing the **Shift**-key, Windows will shoot down with no confirming (If you don't use the Menu as Windows-Shell: **Ctrl + Shift**)

NEW: Improve the customising of the Icons for the Menuitems. The corresponding dialogue functions like the dialogue in the Program-Manager. Now you can also customise the Icons for **Submenus** and **Drive Directory** and the Icons will not be discarded, if you change the icon-size.

If you edit the INI-file by hand you can also declare two Icons in one library for the selected and the unselected state.

NEW: Implemented 'Drag'n Drop-Panels'. This are sticked Menus, which shows only the Icons and not the describing-text. For hints, how to use this Drag'n Drop-Panels look in the help-file for **Drag'n-Drop-Panel**.

NEW: Enhanced the functionality of the Submenu **Tasks**. If you push the **Shift**-key while opening this submenu, also the hidden programmes are listed. If you hold the **Ctrl**-key pressed, while selecting one of the Tasks-Menuitems, the corresponding program will be closed.

NEW: The internal command **MNU\_WIN\_REBOOT** ends Windows and restart it. If you push the **Shift**-key while this, the computer will be rebooted. You can also run a not Windows compatible DOS program with this option. Fore more see **Running a Windows incompatible DOS Program** in the help.

FIXED: If RRKMenu runs as Shell, the option **Save DT**. wasn't enabled. This is fixed.

21.11.1993

### **RRKMenü Version 1.0g(erman) + english Additional**

(only published for registrated Users).

FIXED fixed a silly bug with the editmenu in sticked menu

FIXED fixed a little bug with case and icon-database. ->Have to change the rrkmenu.ini all mnu\_.. to MNU\_... Attention this is important! The lower case (in the old INI-files) will not function any more!

## History of RRKMenu 1.1

- NEW I changed the menuitems gener. Properties to Configuration. I think it is now clearer.
  - NEW Reorganised the dialogues (Associate is via Configuration accessible)
  - NEW I add hotkeys to get Submenus by pressing a Ctrl-char while the menu is popped up (select Configuration and than Hotkeys. This dialogue is quite similar to the dialogue Associate)
  - NEW In the Edit-dialogue you can select an Icon (it will have only an effect in the current icon-size). Alternative you can copy an Icon c:\yourdir\program.ico to represent the corresponding program c:\yourdir\program.exe.
  - NEW Add the possibility to popup a menu at the background of the desktop and on the window-title bars (press the Ctrl while selecting Only desktop and look at the icon representation of the menuitem).
  - NEW If you normally use the middle mousebutton to popup the menu, you can declare a second (sub-)menu for the right mousebutton (add the name of the submenu to SpezialRightMouseButton=My SubMenu in the section [General Properties]). This menu will be popped up with the right mousebutton, if the mousebutton is over the background of the desktop or over a title bar of a window.
  - NEW make the iconic representation of Programm-Manager Items a little bit more safety.
  - NEW Add a new style beta (select F2, or F5 and gen. Props and look at Style. (Notice that a change to or from the style standard will delete all icon-database and the menu will be slow at the first time it analyses the Exe-files and extract the icon-information.)
  - NEW: If you press the **Ctrl**-key while **Menu Setup / End**, you can also shut down Windows, when RRKMenu is not running as shell.
  - NEW: Created a new Style for the Menus called **Beta**, and add for this (and the Styles **Norman** and **OpenLook**) an dialog-option.
  - NEW: Supports File-Sharing in a LAN. Now you can declare in the command line of RRKMENU.EXE the directory, where the RRKMENU.INI and the Icon-database will be found.
  - NEU: If you running RRKMenu as Windows Shell, you can save all open windows (Programmes) with their detailed Positions. If you restart Windows all application will be restored in the same Position on the Desktop. (If you push **Ctrl** while selecting the last Menuitem of the Submenu **Tasks** the windows will not be restored).
- Of course (!?) the Menu can't figure out, which documents are loaded in the applications, the document-based applications will be restored empty.

3.11.1993

### **RRKMenü 0.9ec Prelease with english dialogues**

- NEW: Created English dialogues, overcrowded with mistakes.
- NEW: Maybe fixed bug in PMG's Icon showing.
- NEW: Fixed bug in SearchingDirs with ä (ae) ß (ss) etc. in directories
- NEW: You can't use a whitespace as the first letter in a filename
- Create Message for User
- Implied Shift & Esc to switch the state in sticked Menus
- NEW: Implied Shift & PgnUP etc. to Rollup in sticked Menus
- NEW: Implied Shift & Arrows to move a sticked Menu
- NEW: Totally reworked MEM handles of 'more...' - Submenus in sticked Menus,

## History of RRKMenu 1.1

because it waste the global HANDLES.

Create a Container-Class (only 64 k)

NEW: Create a better Exception-handler for global and local MEM

NEW: Add the Interface for RealMove

NEW: Rewrite the Mem-Management for IdxInRAM using my new Container-Class.

Now I don't need more GlobalHandels, (excepting the code segments) than Program-Manager!

Also, the menu (Icons) becomes faster.

FIXED: Fixed a bug with SubMenus in ff-Menus (normal and sticked Menus)

NEW: Implied Alt + Pause hook

Now the Menu can used without a mouse.

FIXED: Restoring sticked Menus (by using 'Save DT.') doesn't break the menu to a ff (more..) menu.

FIXED: The interface to declare Documents isn't very safe. (it's more alpha then beta)

BETA: Give the program 1k more stack, because sometimes by registrating the Icons in the database the stacks crashes.

NEW: Write the english help

NEW: Fixed the bug with 'more...' restored at the starting of Program

NEW: Improve the first painting of sticked menus

1. they are only painted one time

2. there is no white space flickering

NEW: The submenus now are first build, when they are open, not when they are selected.

This improves the key-support (all the huge, highlighted but not opened Submenus aren't initialised).

NEW: I add a new Look of the menus (OWL\_LIKE=1)

21.10.1993

### **RRKMenu 0.9 Prelease**

#### QUESTIONS TO THE PUBLIC

- Somebody want custimising font-colour in the menu?
- Do anybody use the option showing the menus in the poor/pure Windows-style (deselect the option Look Like /Graphic)? I think about it, to throw out this option.
- Do anybody use the option to declare a directory to the RRKMENU.INI and the Icon-database in the command line of RRKMENU.EXE?

#### QUESTIONS TO WIN-PROGRAMMMER

- Do anybody have a hint, how to subclassing the menus to create gray Menubackgrounds? I only know the way via SetSysColors(...), but this effects all menus.
- How to get rid of the Submenu-marks (the small black triangles) in Ownerdrawn Menus?!!!
- Verify, that the combination Alt + Pause

## History of RRKMenu 1.1

Is not used by other programmes? Any experience???

- HighlightMenu() doesn't function in Submenus!!! (created with CreatePopupMenu())

Any Idea???

- Underlined Chars are not supported in ownerdrawed Menus!!!!

Any Idea???

- I often use CreatePopupMenu(), AppendMenu(...) and DeleteMenu(...) with the style MF\_OWNERDRAW. Sometimes, not regular, these functions corrupt other Menus. F.e. the Menu 'File' in the notepad is not assessable or this Submenu gets an enormous width. Obvious sometimes the function CreatePopupMenu() not deliver an individual HMENU. I only can trap the receive of the global Systemmenu-handle, because it has a static HMENU (0x2058 and 0x006c are constant HMENUs).

Any experience or any hint?

- I think about to implement an interface to New Menus for Windows. There might be two ways:
  1. Calling an DLL, with the name of the DLL, the name of the function and the parameters in the command line.
  2. Context-sensitive popup Menus. Every time the cursor is over a defined window-class a special menu can be popued (similar to the Edit-Menu of a sticked Menu realised in RRKMenu). In the command lines of this menu you could define WM\_-messages to send to this window. I might be also possible to define special menus for standard-window-classes, like edit-boxes, to manipulate this windows via SendMessage(). This way is thinkable to enhance Windows functionally and comfort.

If you are interested to join to this project, please contact me.