













## Index


-  [Important / Getting Started](#)
-  [Introduction](#)
-  [Quick Tips](#)
  
-  [Keyboard](#)
  
-  [An Example Menu](#)
-  [How To Do](#)
-  [Miscellaneous](#)
  
-  [Internal commands](#)
-  [The API](#)
  
-  [The Dialogues](#)
-  [Bugs, Limits, Help](#)

## Important / Getting Started

 [Index](#)

 [Important - read at first](#)

 [Thanks to...](#)

 [What You have to read / know](#)

## Introduction



[Index](#)



[About the Author and The Program](#)



[Installation](#)





[Introduction to the Features](#)


## Installation


 [Index](#)


 [Introduction](#)

 [Requirements](#)


 [Installation of the Menu](#)

 [Install the Menu in a Net environment](#)

 [The Files](#)


 [The Infile](#)


## The Inifile

 [Index](#)


 [Introduction](#)

 [Installation](#)


 [General Properties](#)


 [Submenus in the INI-File](#)


## Introduction to the Features

 [Index](#)

 [Introduction](#)


 [Preliminary Remark](#)

 [Basic functions](#)


 [The features](#)


## Quick Tips


 [Index](#)


 [What is it?](#)

### How to do:

 [Call the Menu](#)

 [Ending The Program](#)

 [Launch Programmes](#)

 [Configure the Menu](#)

## The Dialogues



[Index](#)



[The Dialogue Configuration](#)



[The Dialogue Menu - General Properties](#)



[The Dialogue Menu - Edit MenuItem](#)



[The Dialogue Select Icon](#)



[The Dialogue - Delete MenuItem](#)



[The Dialogue - Delete a Submenu](#)



[The Dialogue Menu - Execution](#)



[The Dialogue - New MenuItem in an empty Submenu](#)



[The Dialogue - Document Association](#)



[The Dialogue ShortCuts for Submenus](#)

[Der Dialog Hotkeys](#)



## Bugs, Limits, Help



[Index](#)



[Known Bugs](#)



[Abnormal Reactions](#)



[It doesn't function - Help!!](#)



[??Beep!!](#)



[Internal Limits](#)

## How To Do



[Index](#)



[Customise the Menu](#)



[Include The Program-Manager](#)



[Using the Sticked Menus](#)




[Using Drag'n Drop](#)



[Using the Menu without a Mouse](#)

## Customise the Menu

 [Index](#)


 [How To Do](#)

 [Customise the Menuitems](#)


 [Miscellaneous](#)


## Customise the Menuitems

 [Index](#)

 [How To Do](#)


 [Customise the Menu](#)


 [Insert a New Menuitem](#)

 [Add a New Menuitem](#)

 [Edit an Existing Menuitem](#)

 [Delete an Existing Menuitem](#)


 [Declare Icons for Non-Windows Programmes](#)


 [Running a Windows incompatible DOS Program](#)

## Using Drag'n Drop


 [Index](#)

 [How To Do](#)

 [Open any File](#)


 [Insert Files as Menuitems](#)


## **Include The Program-Manager**

 [Index](#)


 [How To Do](#)


 [The Program-Manager as Menu](#)


 [A single Program-Group as Menu](#)


 [Edit Menuitems of a Program-Group](#)


## Miscellaneous

 [Index](#)


 [How To Do](#)


 [Customise the Menu](#)


 [Hide the Program Icon](#)


 [Only Desktop](#)


 [Menus with a 3-D-Effect](#)

 [Choosing the Menu-Font](#)

 [Use Icons in the Menus](#)


 [Change the look of the Menu](#)

 [Starting Options](#)


 [Using several INI-files](#)


 [Using ShortCuts to Submenus](#)


 [Using Hotkeys](#)


 [Using the middle Mousebutton](#)

 [Use the right Mousebutton with a Second \(Sub-\)Menu](#)

 [Using a Submenu as a Desktop-Panel](#)


 [Using Context-Sensitiv Menus](#)


 [Improve the Speed of the Menu](#)


 [Avoid Conflicts with other Programmes](#)


## Using the Sticked Menu


 [Index](#)


 [How To Do](#)


 [Stick Menu on the Desktop](#)


 [Move the sticked Menu](#)


 [Minimise the Sticked Menu](#)


 [Hold sticked Menu on the Top of the Desktop](#)

 [Popup the local Editmenu in sticked Menu](#)

 [Arrange all sticked Menu on the Desktop](#)

 [Refresh the sticked Menu](#)

 [Using a Submenu as a Desktop-Panel](#)

 [Stick normal Drop-Down-Menus with a Pin](#)



## An Example Menu



[Index](#)



[An Example Menu](#)



[A Submenu](#)



[A Searching Directory](#)



[A Program Manager Group](#)



[The Program Manager](#)



[A Drive Directory 1](#)



[A Drive Directory 2](#)









[The Menu - Menu Setup](#)



[The Menu Tasks](#)

## The API

 [Index](#)

-  [Internal Commands](#)
-  [Call a DLL-Function](#)
-  [Simulate keyboard actions with a MenuItem](#)
-  [Popup a Menu with an extern Program](#)
-  [Associate a Window-Class with a Menu](#)
-  [Send Messages to a Window](#)

# New Menus for Windows 1.1

© 1993 Roger René Kommer  
Dörnbergstr.11  
D - 34119 Kassel  
Voice. (++ 49 561) 780 150  
EMail: roger@architektur.uni-kassel.de



## Attention

Reading this help still can damage your good connections to the English language!!


This software and the accompanying files are distributed "as is" and without any warranties.  
The user must assume the entire risk of using the program.

For more, see: [Important - read at first](#)





### Important

-  [Important - read at first](#)
-  [Thanks to...](#)





### Getting Started

-  [What you have to read / know](#)



### Introduction

-  [About the Author and the Program](#)
-  [Preliminary Remark](#)
-  [Basic functions](#)
-  [The features](#)


### Installation

-  [Requirements](#)
-  [Installation of the Menu](#)
-  [Install the Menu in a Net environment](#)
-  [The Files](#)




### The Inifile


-  [General Properties](#)
-  [Submenus in the INI-File](#)

### Quick Tips

-  [What is it?](#)


How to do:

-  [Call the Menu](#)
-  [Ending The Program](#)
-  [Launch Programmes](#)

 [Configure the Menu](#)

## Keyboard and internal commands

 [Keyboard](#)

 [Internal commands](#)

## An Example Menu

 [An Example Menu](#)

 [A Submenu](#)


 [A Searching Directory](#)

 [A Program Manager Group](#)

 [The Program Manager](#)

 [A Drive Directory 1](#)


 [A Drive Directory 2](#)

 [The Menu - Menu Setup](#)


 [The Menu Tasks](#)

## Bugs, Limits, Help

 [Known Bugs](#)

 [Abnormal Reactions](#)


 [It doesn't function - Help!!](#)

 [??Beep!!](#)

 [Internal Limits](#)

## How To Do


### Customise the Menuitems


 [Insert a New Menuitem](#)

 [Add a New Menuitem](#)

 [Edit an Existing Menuitem](#)

 [Delete an Existing Menuitem](#)

 [Declare Icons for Non-Windows Programmes](#)

 [Running a Windows incompatible DOS Program](#)


### Using Drag'n Drop


 [Open any File](#)

 [Insert Files as Menuitems](#)

### Include The Program-Manager


 [The Program-Manager as Menu](#)

 [A single Program-Group as Menu](#)

 [Edit Menuitems of a Program-Group](#)


### Miscellaneous

 [Hide the Program Icon](#)

 [only Desktop](#)











 [Menus with a 3-D-Effect](#)

 [Choosing the Menu-Font](#)










 [Use Icons in the Menus](#)

 [Change the style of the Menu](#)







 [Starting Options](#)

-  [Using Hotkeys](#)
-  [Using several INI-files](#)
-  [Using ShortCuts to Submenus](#)
-  [Using Hotkeys](#)
-  [Using the middle Mousebutton](#)
-  [Use the right Mousebutton with a Second \(Sub-\)Menu](#)
-  [Using a Submenu as a Desktop-Panel](#)
-  [Using Context-Sensitiv Menus](#)
-  [Improve the Speed of the Menu](#)
-  [Avoid Conflicts with other Programmes](#)

### **Using the Sticked Menus**

-  [Stick Menus on the Desktop](#)
  -  [Move the sticked Menus](#)
  -  [Minimise the Sticked Menus](#)
  -  [Hold sticked Menus on the Top of the Desktop](#)
  -  [Popup the local Editmenu in sticked Menus](#)
  -  [Arrange all sticked Menu on the Desktop](#)
  -  [Refresh the sticked Menus](#)
  -  [Using a Submenu as a Desktop-Panel](#)
-  [Stick normal Drop-Down-Menus with a Pin](#)

### **The API**

-  [Internal Commands](#)
-  [Call a DLL-Function](#)
-  [Simulate keyboard actions with a MenuItem](#)
-  [Popup a Menu with an extern Program](#)
-  [Assoziate a Window-Class with a Menu](#)
-  [Send Messages to a Window](#)

## About the Author and The Program

My first experiences in programming I made with macros in Word and Excel two years ago, without any imagination, that I would ever look closer to Computers things. (Normally I busy with the philosophical problems of the Aesthetic of Architecture and Music, and by the way I study Architecture.)

More by accident than by willing I contact in autumn 1992 with Visual-Basic, amazed, how simple programming for Windows can be.

At this time first I had the idea for a Menu, popped up with the right mousebutton, to launch applications - the beginning of my career as a 'leisure time programmer'.

Until the end of the year 1992 I create a version of the Menu in Visual Basic.

This first version supports Submenus and Icons (but not with text and Icons).

But the program could not satisfy me, because it needed a lot of System resources.

How easy Visual Basic might be, it is not usable to realise a popup menu.

The standard in programming seems to be C / C++ (Nothing against to Pascal, really!). In spring 1993 I got a C / C++ compiler, never heard something about Pointers, Classes, sortable trees, or Callback-functions and Windows-Messages before; First to learn the language, and second - because it is boring writing only hello world Programmes - to write this Program: my first and single non-trivial program. So I do.

1 1/2 month ago I got an access to Internet, and I looked for other Programmes, which has the same idea. I saw, the Menu is - seeing it at the historical aspect - a 'Me too' or better 'Me more' program. (Some of these Programmes - better: all these, I know - have one fundamental problem: They load the Menu and all Submenus at the start of the program. If you want configure several Submenus, it worsts a lot of System resources - and it doesn't matter if you have 32 MB RAM installed at your Computer, the System resources are absolutely limited under Windows. My Menu generates the Submenu first, when you open the Submenu, and frees all System resources by clicking on the desktop.)

Until this time, I assess my Program only as a fingering-exercise to learn C / C++ and the Windows-API, don't think might be very interesting to public. But I saw, other programmers are also not wizards.

I think with the version 1.1 the New Menus for Windows is not only a 'Me more' program (excepting my help-files!), because increasing quantity of essential and detailed features becomes to a new quality. I hope it's usable for you, and you have fun with it.

### Used Literature:

Besides the documentation of Borland C / C++ 3.1 and parts of the Windows SDK I used following literature:

- Schildt, Herbert; C - Befehlsbibliothek; Hamburg 1987 McGraw-Hill Book Company
- Petzold, Charles; Programmierung unter Microsoft Windows 3.1 (Programming Windows); München 1992 Microsoft Press
- Schulman, Andrew; Maxley, David; Pietrek, Matt; Undocumented Windows; Bonn (...) 1992 Addison-Wesley

### 'Stolen' Ideas:

- 'Open Look' by Open Look Foundation  
The Pin to stick Menus on the desktop.
- 'Corel Draw 3' by Corel Corporation  
The Roll Up Button.
- 'Informant' by James S. Straub  
The idea, that you can simulate a movement of windows in 'real Time' (like NEXT). I improved this simulation through a Message Loop in WM\_MOUSEMOVE to restore the uncovered Regions. But it is still flickering.



## Preliminary Remark

Why using this Program?

It is a question of taste, not really a question of needing.

The Program Manager of Windows have several aspects, which can't satisfy the (power-?)user.

- You can't create subgroups in the Program Groups.
- Often you have to fish the Program Manager out of the overcrowded Desktop.
- It doesn't to be worth to install Documents of your actual Projects in the Program Manager.

Enough of lamentations! How I tried to compensate this faults, see: [Basic function of the Menu](#)



## Basic functions of the Menu

Some other Graphic User Interfaces has not a Program Manager, known by Windows. They use a menu to launch programmes. This menu normally appears by clicking on the desktop.

The model for my Menu is the Desktop Menu of OpenLook (may recognisable by the pin).

After popoped the Menu with the right mousebutton, you can start programmes or open documents.

Similar to Program Groups of the Program Manager you can group all the programmes in Submenus:

Often used programmes and documents in Submenus and seldom used programmes and documents in SubSubSub...Menus.

For an overlook to the important features of the Menu see: [The features of the Menu](#).

## The features of the Menu

- Permanently only 1 % - 2 % of system resources needed (Program Manager ~ 4 - 5 %).
  - Permanently only ca. ~25 KB non-discardable RAM needed (Program Manager ~16 KB. But in the time of virtual memory this might be more theoretical. If there is enough RAM, the Menu allocates ~140 KB, the Program Manager ~190 KB).
  - A free customisable Menu, with Submenus as many as you like.
  - Supports the right as well as the middle mousebutton.
  - You can use the Menu also without a mouse.)
  - Shows automatically the icons of programmes and documents in the menus in 3 possible sizes.
  - Free customisable Menu-font.
  - Free customisable Menu-Icons.
  - ShortCuts for Submenus
  - HotKeys to launch programmes
  
  - Supports following generic Menuitems
    - Supports ambiguous filenames (like `c:\mydir\*.txt`), so you can list all existent files matching a Searching Directory
    - You can include a little generic 'File Manager' in your Menu, in which all Subdirectories of your harddisk showed by Submenus, and all programmes and document of a directory are listed.
    - You can include the Program Manager or single Program Groups in your Menu.
    - Submenus, which becomes too long fitting on the desktop, will break and continued in a new Submenu called More... (It was very hard to realise this. The obvious things are the most complicated.)
    - A Task-List, to switch quickly to an application.
  
  - You can stick the Menus with the pin on the desktop.
    - It is possible to hold the sticked Menus on the top of the desktop
    - Minimise the sticked Menu with Roll-Up-Button.
    - Supports Drag and Drop with the File Manager to install Files as Menuitems in a sticked Menu.
    - Supports Drag and Drop with the File Manager to open any File with a program, installed in a sticked Menu.
  
  - You can also stick normal Drop-Down Menus of Win-programmes with a pin.
  
  - Context-Sensitive Menu enhances the funktionality of Windows controls.
- (I can't describe all features of the Menu at this place. For more please refer to chapter **How to do** in the Index window and to the file ADDONS.WRI. More functions are also described in the chapter The API

## Keyboard

The keys to configure the Menu only function, if you have popuped the Menu and highlighted a MenuItem - respectively a sticked Menu has the focus.

If you haven't a mouse look at Using the Menu without a Mouse.

**F1** shows the Help

**F2** calls a dialogue to change the General Properties

**F5** calls a dialogue to Configure the Menu and edit a MenuItem

**F6** Edit a MenuItem

**F7** Add a MenuItem

**Del** Delete a MenuItem

**Ins** or **F4** Insert a MenuItem

**F9** or **4 on the Numpad**:

Arrange all sticked Menus left on the Desktop

**F11** or **8 on the Numpad**:

Arrange all sticked Menus left on the Desktop

**F12** Refresh a sticked Menu.

**Alt + Pause** to popup the Menu by keyboard

### Ctrl and Shift

- while selecting a MenuItem:  
to change the Commandline before Execution
- If don't use the Menu as Windows-Shell, you can also quit Windows with the MenuItem **End**. If you push the **Ctrl**-key while selecting this menuItem, Windows will shoot down. If you push the **Ctrl**- and the **Shift**-key, Windows will shoot down without a confirmation.
- If you open the Submenu **Tasks** while holding the **Shift**-key down, also the hidden windows will list in the Submenu.  
If you click on a MenuItem of the Submenu Tasks and press the **Ctrl**-key, the corresponding program will be close.
- If you open a Submenu and push the **Ctrl**-key, all Submenus will be closed and the Submenu you opened will showed as the 'Main-Menu'.
- If you click on a MenuItem of a sticked Menu with the **Shift**-key, the Menu will roll up after executing the Menuitems command line.
- If the Menu starts and you press **Shift**-key, Wins DT (saved with **Save Wins DT** or the internal command MNU\_SDT) will not restored. Also - if the Menu runs as Shell - the items in Autostart - will not executed.



## **Document**

A Document is a file associated with a program, declared by the file-extension.  
For example: A File with the file extension `.wri` is a Write - Document.

You can associate Files with Programmes in the dialogue [Dokument -Association](#).

### **Sticked Menu**

You can stick the Submenus on the desktop by clicking on the green Pin. Indeed, it isn't really a Menu, but it reacts quit similar to a menu.

In contrast to normal Menus, the sticked Menus don't disappeared by clicking on the desktop. You can move the sticked Menus by dragging it on the head of the Menu; You can roll it up and down with the red / green arrow button; With another click on the pin you can hold the Menu at the top of the desktop and you can close it by a 3th click on the pin.

**More ...**



The menu supports an automatically 'menu break'. If a Submenu tends to become too long, at the end of the menu a Submenu called **more ...** is appended. This Submenu contains the rest of the Menuitems.

## Highlight Menuitems to edit a Menuitem

### In normal, non-sticked Menus

- If you want to edit a Menuitem, at first you have to popuped the Submenu, by pressing the right (respectively the middle) Mousebutton.
- Secondly you have to highlighted the Menuitem with the left Mousebutton.
- **While** highlighting the Menuitem, you can use one of the Edit keys (f. e. F5).
- After the Menu is disappeared an a dialogue appears, you can release the left Mousebutton.

### In sticked Menus

*Either:*

- Highlight a Menuitem an release the Mousebotton.
- Press an Edit keys (f. e. F5).

*or:*

- In sticked Menus you have only to click with the right (respectively with middle) Mousebutton on the Menuitem to edit.
- In the popu



## Default Icons



General program (Windows or DOS), without an own Icon.



Document of a program, associated with a program without an own Icon.



A COM-Program (DOS-Program).



A DOS-Batch-File.



A DOS-Program with a PIF-file.

### **Popup the Menu**

- If you have select the option **only Desktop** in the Submenu **Menu Setup**, you have to move the mouse cursor to a place, where the background of the desktop is visible.
- Push the right (respectively the middle) Mousebutton.

*or:*

- Push the **Alt** and the **Pause** - key.

### **Close the Menu:**

- Click anywhere on the desktop.

*or:*

- Push one ore several times the **Esc**-key.

## **Menuitem**

A Menuitem conventional contains a command like 'File - Open'.

In the Menu normally a Program will be started by selecting a Menuitem.

Some Menuitems contains Submenus (cascading Menus).

**Iconfile**

Files with the fileextension ICO (sometimes also ICN) contains one Windows Icon. Often also files with the file extension DLL or EXE contains one or more Icons.

Some Icons can be found in the files PROGMAN.EXE and MORICONS.DLL in the Windows directory.

### **Drop-Down Menus**

Drop-Down Menus are the normal Menus of a Windows-program with a menu bar (File, Edit, and so on). If you open a Drop-Down Menu with the right mousebutton, you get a graphic Menu with a pin to stick it on the desktop.

See also: [Stick normal Drop-Down-Menus with a Pin.](#)

## Stick Menus on the Desktop

- Click in a popuped (Sub-)Menu on the green pin.
- To hold the sticked Menu always on the top of the Desktop, click a second time on the green pin, or press the **Shift - Esc** - keys.
- To close the sticked Menu, click a third time on the green pin, or press the **Shift - Esc** - keys.

### Execute a Menuitem of a sticked Menu

A click on a Menuitem the corresponding Program will be start, respectively the Submenu will be open. In contrast to normal Menus a Submenu will first open, if the mousebutton is released.

If press the **Shift**-key while clicking on the Menuitem, the sticked Menu will be rolled up after executing the command line.

If a Menuitem of the sticked Menu is already marked you only have to double-click on the Title bar of the sticked Menu. This also functions, if the Menuitem is a Submenu, or if the sticked Menu is rolled up.

### **move the sticked Menu**

- To move a sticked Menu, drag the Menu with the left mousebutton on the Headline of the Menu, or use the **Shift** and the **arrow**-keys.

See also: Arrange all sticked Menus

### **Minimise the Sticked Menus**

- Click with the left mousebutton on the Roll-Up-Button.

*or:*

- Push the **Shift** and the **PgnDown** - Key

### **To restore the size of sticked Menu:**

- Click with the left mousebutton on the Roll-Up-Button.

*or:*

- Push the **Shift** and the **PgnUp** - Key

See also: Arrange all sticked Menus

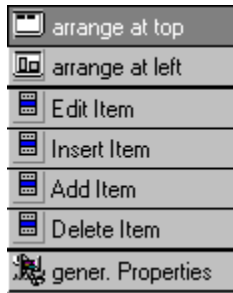


## Popup the local Editmenu in sticked Menu

If the mousecursor is over a sticked Menu and you push the right (middle) mousebutton, a local Editmenu is popped up.

If you want to edit a MenuItem in a sticked Menu, you can click with the right (middle) mousebutton on the MenuItem, and select in the Editmenu the edit-command.

(Hint: Click with the left mousebutton on the Menuitems.)



## arrange at top



Minimises all sticked Menus and orders them on the top of the desktop.  
If an sticked Menu is active, you can also use the **8** on the Numpad or the **F11**-key.

## arrange at left



Minimises all sticked Menus and orders them at the left side of the desktop.  
If an sticked Menu is active, you can also use the **4** of the Numpad or the **F9**-key.

## Edit Item



If you want to edit (change) a MenuItem in a sticked Menu, you can click with the right (middle) mousebutton on this MenuItem. Select **Edit Item**.

In the Edit dialogue you can change the properties of the MenuItem.

See also: Edit an Existing MenuItem

## Insert Item



If you want to insert a new MenuItem in a sticked Menu, you can click with the right (middle) mousebutton on the MenuItem, which marks the point to insert. Select **Insert Item**. In the Edit dialogue you can declare the properties of the new MenuItem.

See also: [Insert a New MenuItem](#)

## Add Item



If you want to add a new MenuItem in a stucked Menu, you can click with the right (middle) mousebutton on the MenuItem, which marks the point to add. Select **Add Item**.  
In the Edit dialogue you can declare the properties of the new MenuItem.

See also: [Add a New MenuItem](#)

## Delete Item



If you want to delete a MenuItem in a stucked Menu, you can click with the right (middle) mousebutton on the MenuItem to delete. Select **Delete Item**, an confirm the dialogue to delete.

See also: [Delete a MenuItem](#)

## Configuration



Calls the dialogue Configuration.



## Arrange all sticked Menu on the Desktop

If you want to arrange all sticked Menus - excepting the Menus only with icons and the sticked Drop-Down Menus - in lines at the top of the desktop:

- select in the local Editmenu **arrange at top**,

*or:*

- push the **F11**-key.

If you want to arrange all sticked Menus - excepting the Menus only with icons and the sticked Drop-Down Menus - in columns at the left of the desktop:

- select in the local Editmenu **arrange at left**,

*or:*

- push the **F9**-key.

## Hold sticked Menus on the Top of the Desktop

It might be desirable to hold a sticked Menu every time in the foreground of the Desktop.

To doing that, click with the left mousebutton on the pin, so the pin looks like: 

You can also use the **Shift-** and **Esc-**key.

If you click once more on the pin, the Menu will be close.

## Refresh the sticked Menu

In different cases it is usable to refresh a sticked Menu:

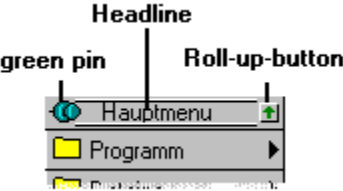
- If you changed the icon- or font size in the [Dialogue for changing general Properties](#)
- If you have a **Searching Directory** in your sticked Menu, and you create or delete a File.
- Last but unfortunately not at least, if there is nonsense in the Menu, or the Menu becomes too long.

To refresh a sticked Menu:

- activate the Menu (there are to thin lines showing, when the sticked Menu has the focus).
- And push the **F12**-key.

The Menu might be disappeared, after pushing F12-Key. In this case, the sticked Menu contains a Drive Directory.

**The title bar of a sticked Menu**



## Requirements

The Menu needs following Hard- and Software:

- Windows 3.1
- A mouse with 2 better 3 mousebuttons  
(It functions also without a mouse)
- A harddisk cache is recommended.

## **Installation of the Menu**

For installation and deinstallation please look in the README.WRI.

Look also: [Das Menü im Netz Installieren](#) and [The Files](#)

## **Entries in the WIN.INI**

The New Menu for Windows doesn't remark its existence in the WIN.INI

## The Ini-file - General Properties

This chapter documents the switches for general Properties in the RRKMENU.INI

### Section [Language]

FirstStart=0

After starting the Menu the first time, it is 0

German=0

1 = German dialogues and help file are used;

0 = English dialogues and help file are used.

(Don't change it by Hand, it's better to reinstall the New Menus for Windows)

### Section [General Properties]

*Item=default*

AsShell=0

0 = Off; 1 = On.

Using the Menu as Shell.

(Don't change this by Hand, it wouldn't function!)

ShowEnhanced=1

0 = Off; 1 = ON.

Shows the Menuitems with a 3-D-effect and customisable Font.

OWL\_LIKE=1

0 Standard; 1 OpenLook; 3 beta.

WithIcons=1

0 = Off; 1 = ON.

Shows the Program Icons in the Menu.

IconSize=16

16, 24 or 32

The Icon size in the Menus.

IDXinRAM=1

0 = Off; 1 = On.

Hold the index file RRKIDX.xxx and parts of the REG.DAT in the RAM.

mMouseButton=0

0 = right Mousebutton; 1 = middle Mousebutton.

The Mousebutton to popup the Menu.

OnlyDesktop=0

0 = Off; 1 = On.

On: The Menu only will popued, when the cursor is over the background of the desktop

NoClient=1

0 = Off; 1 = On.

If OnlyDesktop=1 and NoClient=1 the menu also can be popued over a title bar of a window.

IconVisible=1

0 = Off; 1 = On.

Shows or hide the program icon on the desktop (the black arrow).

SaveDT=1

0 = Off; 1 = On.

Saves and restore the sticked Menus on the desktop by ending the Menu, and restore them by restarting the Menu.

KeySub=1

0 = Off; 1 = On.

Using **Alt + Pause** Hook, to popup the Menu and the Hotkeys.

HookMenus=1

0 = Off; 1 = On.

Normal Drop-Down-Menus of a window application are selectable with the right mouse button.

ClassHook

0 = Off; 1 = On.



Context Sensitive Menus are available.

RealMove=1

0 = Off; 1 = On.

Simulate a movement of the stucked Menu in 'Real-time', known by the NEXT desktop.

mnu\_sdt=1

0 = Off; 1 = On.

Will only has an effect, if the Menu is the Windows Shell.

Controls if a menuitem in the Submenu **Tasks** called **Save Wins DT** is appended. With this Menuitem you can save the positions of all Windows are on the desktop. If the Menu (and Windows) restart, all Windows (programmes) are restored. Unfortunately the documents can't be reloaded.

If you press **Ctrl** while selecting this Menuitem, all positions will be 'deleted'.

SpezialRightMouseButton=(Submenu Name)

Defines a Submenu, you can popup with the right mousebutton over the background of the desktop or over the title bar of a windows.

(This only function, if you normally use the middle mousebutton to popup the Menu.)

OldColorNum=0

It's just to remind the Program Manager to restore it's icons if you changed the number of colors.  
(Don't change this by hand!!!!)

MinResource=7

0 = Off; 1 = ON.

Minimal left of resources  
(Only used for debugging.)

Zeit=50

General Time Variable  
(Only used for debugging.)

### **[MenuFont]**

Name=MS Sans Serif

Size=-11

Weight=400

Width=0

Italic=0

Underlined=0

(Don't change this option by hand)

This Sections may created:

### **[Wins\_DT]**

Stores a list of applications with their coordinates.

### **[ShortCuts]**

A=My Submenu

Ctrl - Char=Name of Submenu.

It's for the ShortCuts to a Submenu.

### **[ProgrammHooks]**

LONG-Number=Command Line

List of defined Hotkeys.

LONG-Zahl: Code of combination of Keys.

Command Line: command line in the same conventions of a Menuitem.

## Submenus in the INI-File

Don't delete the sections **[MainMenu]**, **[Menu Setup]**, **[Konfig\_ResMenu]**

This chapter documents the entries for Submenus and Menuitems in the RRKMENU.INI. Indeed, You don't need to edit the RRKMENU.INI by hand. All is editable by dialogues. But I think, it is a standard to document the important files.

The RRKMENU.INI has the normal format of Windows INI files. Using my own format might be improve the speed of Menu for a few milliseconds, but for the disadvantage of human unreadable file format it isn't worth.

Each Submenu are marked with brackets "[" "]". Below the Menuitems are listed.

Following Submenus are reserved:

[MainMenu]

The (Sub)Menu shows at first.

[Menu Setup]

There are some items to configure the Menu.

[Konfig\_ResMenu]

The local Editmenu in sticked Menus.

[Autostart]

Running the Menu as Windows Shell, all Items listed will be started at the beginning of Windows.  
(You can create this Submenu in the same way, as other Submenus.)

[Wins\_DT]

The Programmes witch saved with the command MNU\_SDT.

[WINCLASSES]

All hooked Windows-Classes are listed here.

[ProgrammHooks]

The hooks to command lines.

[Menu\_Desktop]

Sticked Menus on the desktop to restore.

### Example:

[My Submenu]

; The name of the Submenu.

0=0

1=0|Paintbrush|pbrush.exe

2=1|Files|NIX

3=2|\*.DOC|C:\D\C\MENU\\*.DOC

4=8|c:\windows|c:\windows

### Characteristics of a Submenu:

The first entry  $0=n$  is not a Menuitem, but a number to describe the characteristics of a Submenu. If the entry  $0=n$  is missing, the default for  $n$  is 0;

The number  $n$  is interpreted as bits:

- 1: The Submenu is showed only with the Icons.
- 2: The Subemenu is showed without the pin to stick the Submenu on the desktop.
- 4: The Submenu is showed without the Icons, only with the text.
- 8: The Submenu is locked and not editable with dalogues.

For a combinations of this characteristics simple add the numbers.

### Grammar of a Menuitem:

MenuitemNumber=Typ of Menuitem|Label|File [Arguments][>Working Directory][<WindowSize][;Iconfile  
Iconpos]

[] = optional

- MenuitemNumber: individual Number in the order of the Menuitems
- Type of Menuitem:
  - 0: normal Menuitem  
The **Label** is showed in the Menu and the **File** will be executed
  - 1: Submenu  
The Label is showed in the Menu as a Submenu. It links to an other Submenu. The File is a Dummy (NIX == Nothing), produced by the Menu.
  - 2: Searching Directory  
The Label is a Dummy, produced by the Menu. All Files are listed in the Submenu, these are matching to **File**.
  - 8: Drive Directory  
The **Label** is showed in the Menu as a Submenu. The **File** is the starting directory; all Subdirectories of this starting directory are generate as Submenus, all Programmes and Documents are showed as Menuitems.
  - 128: vertical Dividing Line. **Label** "-" and **Filename** "x-"; x is a individual number.
  - 256: horizontal Dividing-Line **Label** "" and **Filename** "x"; x is a individual number.
  - 8193: Program Groups  
Generate a Submenu with the Programmes installed in a Windows Program Manager Group.
- Label: see **Type of Menuitem**
- File: see **Type of Menuitem**  
*reserved expressions:*  
For more informations, look at Internal Commands
- Arguments:  
The arguments of a Program  
Only used if the Type of Menuitem == 0  
Default is no argument
- Working Directory  
default is the directory of the program
- WindowSize:  
The size of starting Program  
0 = hidden; 1=normal; 2=minimized; 3=maximized  
Only used if the Type of Menuitem == 0  
Default is the default of the program.
- Iconfile Iconpos:  
If the Icon of the menuitem is explizit declared, the file which contains the Icon and the IconID

(Comment: "Unsinn" is German and means nonsense. Ignore this entries.)

## The Files

Please refer to README.WRI for the actual list of the files.

Following files may created in the directory of RRKMENU.EXE:

RRKDAT.xxx

RRKIDX.xxx

If You use the feature to show the program and document Icons, the Menu extract the Icons from the Programmes and write this information in two database files.

The fileextentions "xxx" are named after a combination of the Icon size and the colour resolution of Your graphic driver.

If You use a colour resolution with two colours (black and white) the Bitmap-file needs 1 Bit for each pixel (.01x) 16 colours needs 4 Bits (.04x) and 256 colours etc. 8 bit (.08x) etc.

If You use an icon size of 16 Pixel the third 'x' is a 1 (.xx1); by using an icon size of 24 Pixel a 2 (.xx2) and for 32 Pixel a 3 (.xx3).

The file RRKIDX.xxx is the index file for RRKDAT.xxx. In last the Icons are stored (Not really a Windows-Icon, res. a Windows- Bitmap, more a Core-Dump of a Bitmap, because real Bitmaps would be to slow. This causes the different xxx's. In the case You use the Icon size of 32 Pixel with a Colour resolution of 16 million colours this file (RRKDAT.243) might be become huge.)

If you change the colour resolution of Windows, or you want to change the Icon size, You can delete the old files. But always delete both files!



## Restart Windows

### Yes

Windows will be end and restart.

If you have choosed the Option **As Shell**, instead of the Program Manager of Windows the Menu will start at the beginning of Windows.

### No

Windows will not restart.

If you have choose the Option **As Shell**, instead of the Program Manager of Windows the Menu will start at the next beginning of Windows.

### Cancel

Our changes for the option **As Shell** will be discarded.

**Sorry; Drive directories, Searching Directories and PM-Groups can't be edited in sticked Menus. Use the Edit-Keys.**

You can't edit Drive Directories, Searching Directories and PM-Groups in sticked Menus. To change the Menuitem popup the Menu, highlight the Menuitem and use the Edit keys.

## **There is no MenuItem selected!**

You have to select a valid MenuItem, which You want to edit. Use the right (middle) mousebutton and click on the MenuItem to edit.



**File-Error - File can't be opened.**

Restart Windows to close all files.

**A Hint: You can only edit the beginning of a Drive Directory!**

You highlighted a Menuitem contains a Drive Directory.

If this Menuitem is not the beginning of the Drive Directory, changes won't have an effect.

**You have to declare a Label and a File!**

Look: [The Edit Dialogue](#)

## **You use a reserved expression!**

The expression as filename "mnu\_"... is reserved. You can't edit Menuitems, containing this expression.

## **Attention! Do you wish to keep the old Submenu?**

If you edit (change) a MenuItem containing a Submenu, you have to choose, the old Submenu should be deleted or preserved.

## **Exit Windows?**

**Yes:**

Exit Windows.

**No:**

Neither Windows nor the Menu will be ended.

**This File-Extension is already associated with a Program! Before making a new Association, detach the old one.**

Look: The Dialogue Document - Association

**You have to declare a Program-File and a File-Extension!**

Look: [The Dialogue Document - Association](#)



**The declared Association is not present!**

Look: [The Dialogue Document - Association](#)



## Known Bugs

*Problem:*

If you are too fast by selecting Submenus in stucked Menus, it can be ignore your mouse-action.

*Solve:*

Keep calm, do the same, but a little bit slower.

*Problem:*

Sometimes (maybe depending on the mouse-driver) the Menu annoy you, popping up, although you don't press the right (middle) Mousebutton.

*Solve:*

Press the left and after that once more the right Mousebutton.

*Problem:*

You use an other tool, with hooking some keys to make some action, if you press one or more special defined keys. If you running New Menus for Windows, this tool may not get in action.

*Solve:*

Disable the option **Alt + Pause - PopUp** in the dialogue General Properties.

*Problem:*

Seldom the Menu can corrupt other normal menus. If you recognise, that other menus are changed in an abnormal way, you must restart Windows.

There is a seriously bug in the Windows-Management of the menus. (CreatePopupMenu(), AppendMenu(...) and DeleteMenu(...) seem sometimes to fail in the depth of the Windows-API by stressing out of the normal efforts. Sometimes calling CreatePopupMenu() I even received the HMENU from the System-Menu!!! (That's a Bug!!!(?)) Any idea?)

*Problem:*

The **vertical Dividing Line** is do not function in a desirable way. Especially if you stick the a Menu, containing a **vertical Dividing Line** the stucked Menu can become to long.

*Solve:*

In this version of New Menus for Windows - if you aren't failed in love with this option - avoid it.

## Abnormal Reactions of the Menu

- *Subject:* Edit keys.  
The Edit keys (F5 etc.) don't function, if a DOS- box is active (maybe depending to the mouse-driver).
- *Subject:* Editing of **Searching Directory**.  
It is not possible to edit Searching Directories in sticked Menus
- *Subject:* Association.  
If you marked the option **Index in RAM** in the Dialogue General Properties, the Menu evaluates the file REG.DAT, otherwise the sections in the WIN.INI are used.  
If you change the section "Extensions" in the WIN.INI by hand, the menu may doesn't recognise these changes.

## It doesn't function - Help!!

***Problem:***

The Menu don't popup by pressing the right Mousebutton.

Try it with the middle Mousebutton or deactivate the Option middle Mousebutton

***or***

You marked the option Only Desktop

Move the Cursor over the Background of the desktop and try it again.

or:

Deactivate the option **Only Desktop**.

## Internal Limits

- Showed Menus and Submenus at one time:  
If you reach the limit of the System resources a Beep will be sound.  
(The Menu frees the System resources, if you clicking on the Desktop.)
- Listed Menuitems in a sorted **Searching Directory** or an **Drive Directory**  
1000 (only including programmes and Documents).
- Maximum of Submenus and Menuitems  
The file RRRMENU.INI must not be bigger than 32 KB.
- Registered (and showed) Icons in the Database file RRRKIDX.xxx  
The file RRRKIDX.xxx must not be bigger than 32 KB.
- Sticked Menus  
50

## **!!BEEP??**

The New Menu for Windows sounds in follow cases:

**You selected a MenuItem and execute it,**  
The Program or document can't be found.

**You have selected a Submenu**  
You reached the limits of the system resources  
The Menu frees the System resources, if you click on the Desktop.

**Otherwise:**  
A general trapped fault - I hope it is not serious.





## **What is it?**

The New Menus for Windows are a Popup Menu, in which you can install programmes and documents to launch.

If you push elsewhere the right mousebutton, this Menu will be popued at the position of your mousecursor.

## Call the Menu

Press the **right mousebutton!**

### **To close the popuped Menu**

Click elsewhere with the left mousebutton on the desktop.

If you push on the green pin, you can stick the Menu on the desktop. If you want to get rid of this sticked Menu, click to times more on the Pin.

## Ending The Program

- Popup the Menu and select **End** in the Submenu **Menu Setup**.
- or:*
- Select **Close** in the Systemmenu of the program-icon (a black arrow).

## Launch Programmes

The Menu is not configured for your needings.

The Menu at first is only configured, with a few Menuitems. I supposed, your Windows is installed at `c:\windows`. In the case, my hypothesis is correct, you can access the Menuitems in the Submenu

**WinProgrammes**. Otherwise you have to change these Menuitems.

I also supposed, you have an disk `c:\`.

(But this configuration really don't exhausts the possibilities of the program.)

If you select a Menuitem in the Submenus or Sub-Submenus **WinProgrammes**, **C:\** or **Progr. Manager** normally a program will start or a Document will open.

## Configure the Menu

Because the wishes of the users and also the possibilities of this program are manifold, I give here only a short hint:

If you have popuped the Menu and highlighted a Menuitem with the left mousebutton, you can push the **F5**-key, to edit, insert, add or delete a Menuitem.

See also Keyboard for more Edit keys  
and read the chapters of How to do in the Index window.

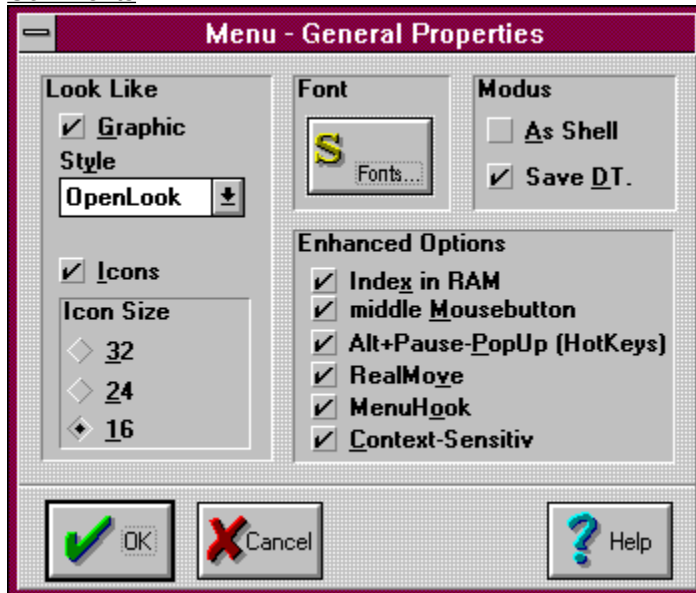
## The Dialogue Menu - General Properties

(Hint: Click on the buttons and fields for more information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



**How to get the Dialogue:**

- Press in the popuped the **F2**-key.

*or:*

- Press in the Dialogue Configuration (f.e. **F5**) **gen. Props.**

**Explanations**

In this dialogue you can change the general Properties of Menu.



## Comments

Nothing

**Index in RAM**

If you haven't seriously problems with your RAM, it is recommended to use this option: The menu gets faster by using this option.

**Middle Mousebutton**

If you have a mouse supporting 3 mousebuttons, it is recommended to use this option. To pop up the menu, you can use the middle mouse-button instead of the right mousebutton.

See also: [Avoid conflicts with other Programmes.](#)

**As Shell**

To use the menu as shell instead of the Program-Manager.

Look also: [Starting Options](#)

**Save DT. (Desktop)**

Saves the sticked Menus on the desktop by ending the menu and restore them by restarting. Submenus build by **Drive Directories** will not be restored.

### **Look Like - Graphical**

The menu has a 3d-effect.

With Fonts you can choose a Font for the Menuitems.

Comment: The sticked Menus always are showed with the 3d-effect.

### **Look Like - Icons**

One of the highlights of the Menu is the feature, to show automatically the Icons of Programmes and Documents in the Menu.

But if you have a slow Computer, you may don't want this option.

Look also The meaning of the Icons

**Iconsize**

The Numbers corresponding to the size of the Icons.

The normal size of the Icons - showed also by the Program-Manager - is 32 pixels. 24 and 16 pixels are smaller.



**Fonts**

A standard dialogue for choosing the menu-font.  
The standard font is MS Sans Serif 8 pt.

**Realmove**

If you have a fast Graphic-System in Our Computer, you can use this option, to simulate a real-time movement of the sticked Menus, known by the NeXT-Systems (Try it, it is just a joke).

### **Alt + Pause Key (Hotkeys)**

Principally the Menu can used without a mouse (on notebooks maybe). Using this option, you can popup the Menu by pressing the Alt and the Pause key. To navigate in the Menu and Submenus, use the Cursors on the keyboard.

See also: [Using the Menu without a Mouse](#)

If you want use [Hotkeys to launch programmes](#) this option must be activated.

**OK**

To confirm the changes in this dialogue.

**Cancel**

Selecting Cancel will undo all changes, except the changes in "Fonts".

## Style

**Standard** is a simple style, but it has the advantage the menuitems are small and the Submenus are build quickly.

The style **OpenLook** is overtaken from a Unix X-Windows GUI.

The style, called **Beta** is just an one-evening-production.

If you change the style to or from **Standard** all icon database files RRKIDX.\* and RRKDAT.\* are deleted.

## **MenüHook**

If this option is on, you can select a drop-down menu of a normal Windows program with the right mousebutton. This Menu has also an pin to stick in on the desktop.

See also: [Stick normal Drop-Down-Menus with a Pin.](#)

**Context Sensitiv**

If this option is on, the context sensitiv popup menus, described in the file ADDONS.WRI are enabled.  
See also: [Using Context-Sensitiv Menus](#).



## The Dialogue - Delete Menuitem

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



The image shows a dialog box titled "Menü Löschen" (Delete Menu). The main question is "Menüpunkt Löschen?" (Delete menu item?). The dialog contains four input fields:

- (Unter-)Menü:** Hauptmenu
- Bezeichnung:** Programm
- Dateiname:** NIX
- Arbeitsverzeichnis:** (empty)

At the bottom, there are three buttons: "OK" (with a green checkmark), "Abbruch" (with a red X), and "Hilfe" (with a blue question mark).

### How to get the Dialogue:

- At first press the **F5**-Key in the popuped Menu and then select **Delete**;
- or.*
- Press the **F4**- or the **Del**-Key in the popuped Menu;
- or.*
- In sticked Menus select the corresponding Menuitems of the Editmenu.

**Explanations**

This Dialogue is to confirm the deleting.

**Comments**

None

**(Sub-)Menu**

The submenu, which contains the actual selected MenuItem, is showed here.

**Label**

The Label of the actual selected MenuItem.

**Filename**

The command line of the actual selected MenuItem.

**Working Directory**

The Working Directory of the actual selected Menuitem.



**OK**

To confirm the deletion of the actual Menuitem.

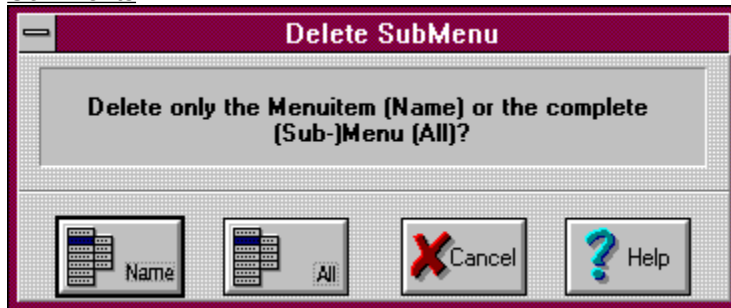
## The Dialogue - Delete a Submenu

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



### How to get the Dialogue:

- If you have selected a popued Menuitem, which contains a Submenu and you press the Del-Key, the Menu asks you, if you want only delete the Menuitem, without deleting the whole Submenu (**Name**), or if you also want to delete the entries of the selected Submenu (**All**).

**Explanations**

None

**Comment**

It is recommended to delete the whole Submenu - conditionally you don't need the Submenu at a other place in your Submenu structure - because the RRKMENU.INI must not became bigger than 32 KB.

## Name



Confirming with **Name**, only the Menuitem, which links to the (Sub-)Submenu in the actual (Sub-)Menu, will be deleted. If you Insert the (Sub-)Submenu in an other (Sub)Menu of your Submenu structure, you can access the old Entries. This might be a way to move whole Submenus.

## All



Confirming with **All**, the whole Submenu will be deleted. This will have the consequence, that other Menuitems with a link to this Submenu will be empty.

## The Dialogue Menu - Edit Menuitem


(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)

**Menu - Edit Menuitem**

**[Sub-] Menu:**  
Programm 


**Label:**  
WINFILE.EXE


**Filename:**  
C:\WINDOWS\WINFILE.EXE


**Working Directory:**


**Hotkey**  
Ctrl+


Menutyp	Menu - Generic	Window-Size
<input checked="" type="checkbox"/> Menuitem	<input type="checkbox"/> Searching Direct.	<input checked="" type="checkbox"/> Normal
<input type="checkbox"/> Submenu	<input type="checkbox"/> Drive Directory	<input type="checkbox"/> Minimized
<input type="checkbox"/> h. Div. Line	<input type="checkbox"/> Progr.-Manager	<input type="checkbox"/> Maximized
<input type="checkbox"/> v. Div. Line	<input type="checkbox"/> Progr. Group	<input type="checkbox"/> hidden

 OK

 Cancel

 Browse...

 Icon..

 Help



### How to get the Dialogue:

There is more than one way:

- At first select in the popuped Menu a Menuitem, than press the **F5**-Key and choose in the dialogue Configuration one of the buttons: **Add**, **Insert** or **Edit**.

*or:*

- At first select in the popuped Menu a Menuitem, than press the **F6**-Key to **edit (change)** a Menuitem.
- ... the **F7**-Key to **add** a new Menuitem
- ... the **F4**- or the **Ins**-Key to **insert** a new Menuitem.

*or:*

- Choose in a sticked Menu the corresponding Menuitem of the local Editmenu.

**Explanations**

In this Dialogue you can insert, add or change a Menuitem.

### **Comments**

In subject with the sticked Menus and the Drag and Drop - method with the File-Manager, you can also add or insert new Menuitems, without needing this dialogue.

Look: [Drag'n Drop: Insert files in a Submenu](#)

**(Sub-) Menu**

The Submenu, which contains the actual selected MenuItem, is showed here.

**Label**

The Label, which is showed as Menuitem in the Menu. The Label must not longer than 30 chars. Internal (As a link to a Submenu) there is no different in case.

**Filename**

Normally a program-file or a Document- to execute by choosing a Menuitem.

("MNU\_"... is reserved for internal command)

### **Working Directory**

You can declare a standard working path for programmes.  
(This will have only an effect for standard **Menuitems**.)

## Menuitem

A normal Menuitem needs a **Label** and a **Filename**.

The **Label** is depending from your imagination. (f.e.: My fine Program)

Use **Browse** for picking a Program- or a Document- File for **Filename**. The trial to execute other Files, than Programmes or Documents are receipted by a Beep.

You can also declare Arguments after a whitespace in **Filename**: f.e.: c:\mydir\myprogr.exe /v /r myfile.dat

In the field **Working Directory** you can declare a path for the program.

Also you can choose the window size of the starting program in **Window-Size**.



## Submenu



A Submenu is a cascading menu.

Choosing a MenuItem, contains a link to a Submenu, this Submenu will be open. The Submenus are defined by their **Labels**. f.e.: Games and More

Intern there are no differences between capital and small letters. A declaration for a **Filename** is not necessary (NIX is 'short' form for "nothing").

Instead of the **Window Size** you can declare the **Style of a Submenu**.

### **horizontal Dividing-Line**

Use this, if you want to structure your Menuitems in several sections.  
You have to accept the suggested entries in **Label** and **Filename**.

### Searching Directory

This option generate Menuitems by using DOS ambiguous file-names.

If you write `c:\mydir\*.txt` in the field **Filename**, in the Submenu all matching textfiles are listed in a alphabetical order. Entries in the field **Label** won't have an effect, because they are replaced by the found filenames (without the path).

You can use **Browse** to select the directory and a file.

(Comment: You can collect your actual project files in a Submenu, stick this Submenu on the desktop, and you doesn't need the Menu 'File - Open' in our application.)

## Drive Directory



(The name may be unfortunately)

includes a little 'File-Manager' in the menu. In a submenu at first all directories and secondly all executable files and documents are listed in a alphabetical order.

In the field **Filename** you have to write the starting directories for example `c:\` or `c:\mydir\mysubdir`.

(After choose this option, you can use **Browse** to chooese the starting path by selecting a Dummy - file.)

In the field **Label** you can use the defaults offers by **Browse** or you can declare your own Label.

### **Window - Size**

The declaration of the Window - Size will have only a effect by using standard **Menuitems**.  
If the MenuItem is a Submenu you can declare the **Style of a Submenu** at this position.

**OK**

After editing the MenuItem and confirming with OK the Menu will be change.

### **vertical Dividing Line**

You can use this item to separate the Menuitems below the vertical Dividing Line in a new column.  
(This feature is still 'Alpha')

You have to accept the suggested entries in **Label** and **Filename**.

## Progr.- Manager



You can include the Program-Manager in the menu with this option.

You must accept the suggested default in the field **Filename**.

You can use the default in the field **Label**, or you can change it.



## Progr.- Group



You can include a single Program-Group from the Program-Manager in the Menu.

Use **Browse** for searching a existing '\*.grp'-file in your Windows-Directory.

In the field **Label** you can use the default, offers by **Browse**, or you can declare your own Label.

(Hint: You can edit the menuitems in "Progr. Manager" and "Progr.-Gruppe" in the normal way. But this option has the disadvantage, that the Icons are showed more slowly.)

**aktual Icon**

The actual Icon of the menuitem is showed here.

**Icon...**

Opens the dialogue [Icons](#) to declare an Icon for the current MenuItem.

See also: [Declare Icons for Non-Windows Programmes](#).

## Hotkey

A Hotkey is a combination of keys, to execute the command line - inclusiv parameters, working directories and window sizes - every time.

Put the cursor in the editbox and press the key combination.

See also: [Using Hotkeys](#) or [The Dialogue Hotkeys](#).

If the actual Menuitem is a Submenu, you can use this editbox to declare a ShortKey to the Submenu. To do this, press the **Ctrl**-key and a char. If this ShortKey is already associated with an other Submenu, the old association will be overwritten.

See also: [Using ShortCuts](#) or [Dialog ShortCuts](#).

## **Submenu - Styles**

### **Normal**

The Submenu and the Menuitems are showed in the default style.

### **o.Icons**

The Menuitems of the Submenu will be showed only with the icons.

### **o.Text**

The Menuitems of the Submenu will be showed only the the describing text.

### **NoPin**

The Submenu will not have a pin to stick it on the desktop.

### **Locked**

You can't edit the Menuitems of the Submenu with dialogues.

## The Dialogue Menu - Execution

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



**How to get the Dialogue:**

You have to press the **Ctrl-** Key while executing a MenuItem.

**Explanations**

You can change the filename and his parameters in dialogue before executing.

It is also a comfortable replacement for the Program-Managers "File - Execute" (or similar, I haven't the English Version of Windows)



**Comments**

None

**Command line**

Here is the Filename of the program or the document.

**Parameter**

Like a filename or flags, are expected by the program.

### **Working Directory**

Some programmes need to start in a special directory, otherwise they can't find their files.

**OK**

After confirming, the program starts, respectively the document will be open.

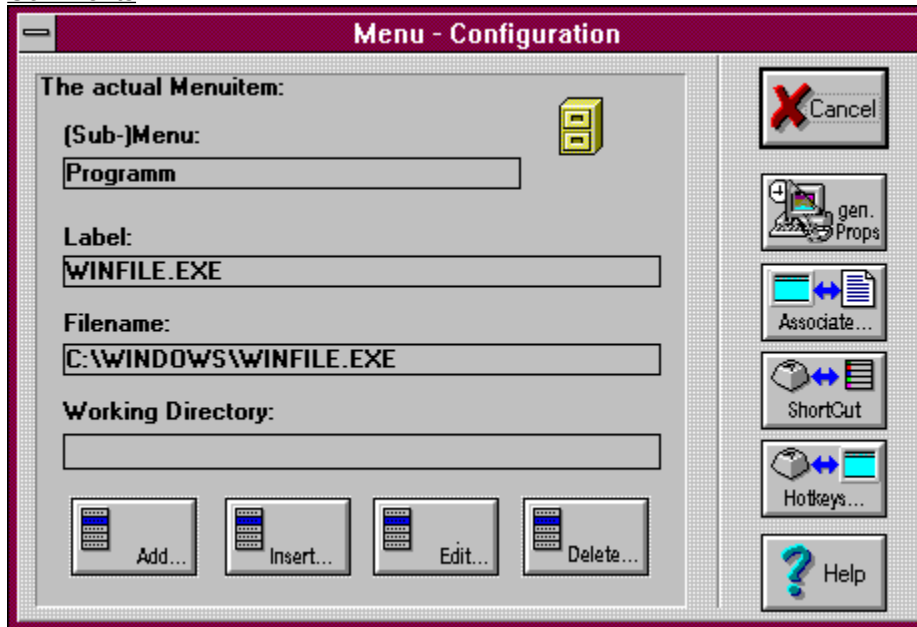
## The Dialogue Menu - Configuration

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



**How to get the Dialogue:**

- Press in the popuped Menu the **F5** Key.

*or:*

- Select in the Submenu **Menu Setup** the item **Configuration**

*or:*

- Select in the Edit-Menu of sticked Menus the item **Configuration**

*or:*

- Select in the system Menu of the program icon the item **Configuration**

(You see, you can't miss it.)

**Explanations**

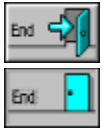
None



**Comments**

None.

## End



Ends the Menu.

If the Menu is running as Windows Shell or if you press the **Ctrl**-key, after confirming Windows will shoot down.

## Edit Menuitem



Opens a Dialogue, to edit the actual Menuitem.

Look: Edit a Menuitem

Dialogue Menu - Edit Menuitem

## Delete Menuitem



Deletes the actual Menuitem.

Look: Delete a Menuitem

Dialogue Menu - Delete a Menuitem

## Add Menuitem



Opens a dialogue, to add a new Menuitem below the position of the actual Menuitem.

Look: Add a Menuitem

Dialogue Menu - Edit a Menuitem

## Insert Menuitem



Opens a dialogue, to insert a new Menuitem above the position of the actual Menuitem.

See: [Insert a Menuitem](#)

[Dialogue Menu - Edit a Menuitem](#)

## General Properties



Opens a dialogue, where general Properties of the Menu can be changed.  
Look: Dialogue General Properties

**(Sub-)Menu**

The Submenu, which contains the actual selected MenuItem, is showed here.



**Label**

The Label of the actual selected MenuItem.

**Filename**

The command line of the actual selected MenuItem.

**Working Directory**

The Working Directory of the actual selected Menuitem.

**Browse:**



This button will be open a standard-dialogue for picking a file.

**Cancel**

To cancel the action.

**Cancel**

To cancel the action.

**Help**

gives Help for the dialogue.

## Associate...



Starts a dialogue to associate Documents (declared by the file extensions) with a program.  
See also: [Dialogue - Document - Association](#)



## **ShortCuts**

Opens the dialogue ShortCuts.

**Hotkeys**

Opens the dialogue Hotkeys.

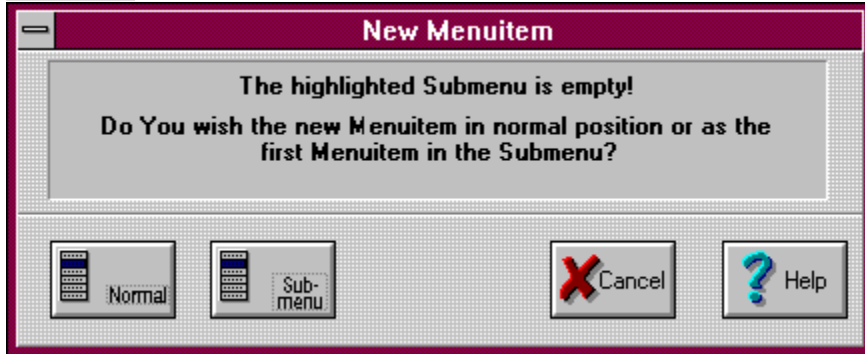
## The Dialogue - New Menuitem in an empty Submenu

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



**How to get the Dialogue:**

If you insert a new MenuItem, and you highlighted an empty Submenu, this dialogue appears.

**Explanations**

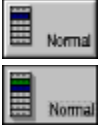
To insert the first MenuItem in an empty Submenu, you have to highlight the empty Submenu and press the Ins-key.

After declare the new MenuItem in the Edit dialogue you can choose, if you want this new MenuItem as the first MenuItem in the empty Submenu.

**Comments**

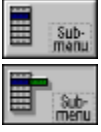
None.

## Normal



The blue Menuitem contains the empty Submenu. If you choose **Normal**, the new Menuitem (green) will be get the position above. The empty Submenu will be still empty.

## Submenu



The blue MenuItem contains the empty Submenu. If you choose **Submenu**, the new MenuItem (green) will become the first MenuItem of the empty Submenu.



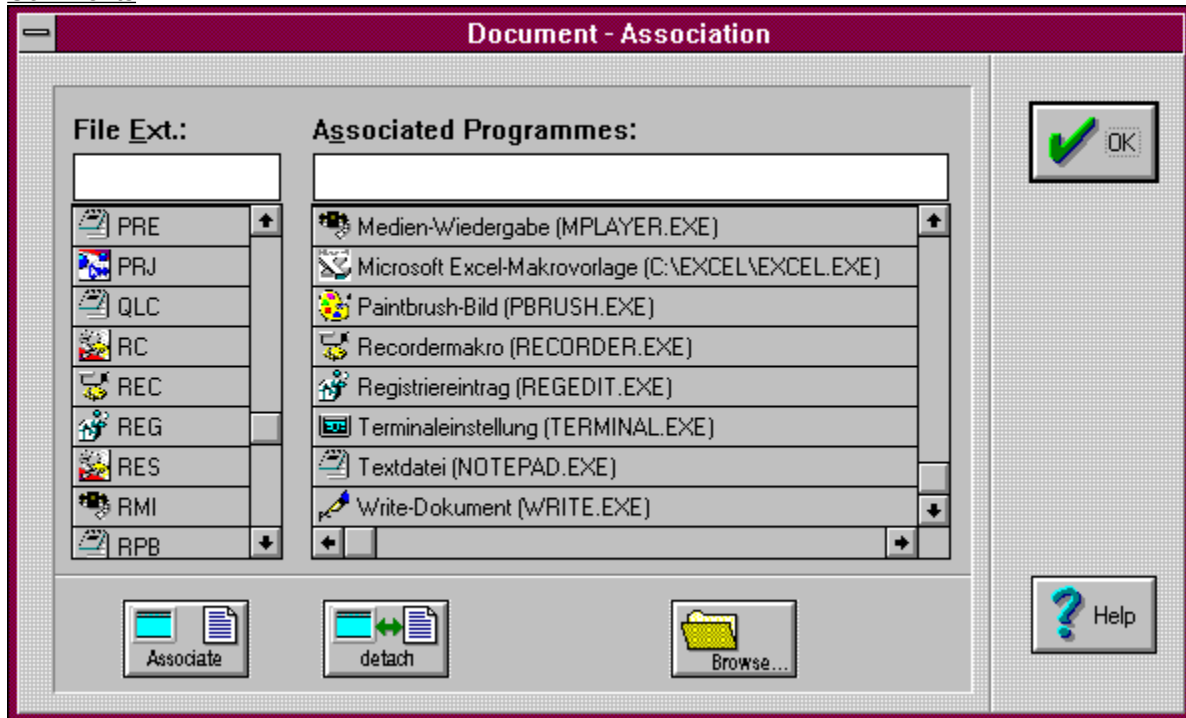
## The Dialogue - Document Association

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



**How to get the Dialogue:**

- Press in the popuped Menu the **F5**-key.

*or:*

- Select in the Systemmenu of the programicon (a black Arrow) **Configuration**.

*or:*

- Select in the Submenu **Menu Setup** the Menuitem **Configuration**.

*or:*

- Press in the popuped the **F5**-key and select in the dialogue **Configuration**.

*or:*

- Push over a sticked Menu the right (middle) Mousebutton and select in the local Editmenu **Configuration**.

**And:**

- Select **Associate**

## **Explanations**

In this dialogue You can declare Windows Documents by associate a fileextention with a program.

How to do:

Define a Document

Undefine a Document

**Comments**

Documents are defined by their file extension. For example: A file with the file extension `.wri` is a Write - Document.

**File Ext. (File Extension)**

In this field You have to insert the File-Extension

### **List of the existing File Extensions**

All File Extensions, defining a Document, are listed.

If You click on one File Extension, in the list **Associated Programmes** the Program associate with the document will be highlighted.

**Associated Programmes**

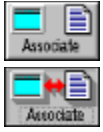
In this field You have to insert the Program.

**List of the associated Programmes**

Here all Programmes listed, are registered in the REG.DAT



## Associate



If You have declared a File Extension in the field **File Ext.** and a program in **Associated Programmes** You can define a Document.

## Detach



If You have declared a File Extension in the field **File Ext.** and a program in **Associated Programmes** You can undefine a Document.

**OK**

To close the dialogue.

### Define a Document

Select in left list a File Extension to detach and push the button **detach**.

### Undefine a Document

- Declare a File Extension in the field **File Ext.**
- Select a Program in the list **Associated Programmes**

*or:*

- Use **Browse**, to pick up a program file.

Querverweis von VKN\_DIALOG\_HELP\_ID auf vkndlg

## Important - read at first

### Copyright:

This software and the documentation is copyrighted.

The files BWCC.DLL and BWCCGER.DLL (german version of BWCC.DLL) are copyrighted by Borland International.

Trademarks are - of course - Trademarks.

### Agreement for Using / Warranty

This software and the accompanying files are distributed "as is" and without warranties as to performance of merchantability or any other warranties whether expressed or implied.

Because of the various hardware and software environments into which RRKMenu may be put, no warranty of fitness for a particular purpose is offered.

The program still have some bugs, see also [Known Bugs](#).

The user must assume the entire risk of using the program.

### Distribution:

This version of RRKMenu may be freely distributed, subject to the following restrictions:

It may be freely used by individuals, educational institutions and charities; use within commercial organisations is not permitted without the written consent of the author. The program may be stored in software archives or posted to Bulletin Board systems, and may be distributed by individuals. Apart from reasonable media and handling costs, no charge may be levied for its distribution.

Is prohibited to distribute the software and/or documentation with other commercial products without prior written permission.

The program should be stored with all the files listed in [The Files](#).

### Support:

A special Support is not offered by the Author.

Reports to bugs and mistakes in the program and the help are welcomed.

### Registration / Feedback

RRKMenu is FreeWare or ShareWare - How do you wish ;- ) - but anyway not public domain.

I think it is neither worth to annoy the user with restrictions (like reminder screens or other limitations) nor to make the user a bad conscience using software illegal. You can 'test' RRKMenu as long as you wish. A different in functionality of RRKMenu between unregistered and registered versions doesn't exist.

But if you want appreciate the work of a 'poor student' with a fee, if you want keeping my nose to the grindstone for next releases, if you want a support regarding RRKMenu and/or if you want the actual version (if there is one), you can register RRKMenu for 30 \$ US or 50 DM. It would be the best, to send it cash by normal mail to my surface adress, because the service charge for checks or bank transfer is over half of the sum! (Yes, they are real gangsters in pinstripe suit {:-( )

Inquiries, comments, and suggestions regarding RRKMenu are also welcomed and can be sent to

Roger René Kommer

Doernbergstr.11

D 34119 Kassel

Germany

Voice. ( ++ 49 561 ) 780 150

E-Mail: [roger@architektur.uni-kassel.de](mailto:roger@architektur.uni-kassel.de)

## What you have to read / know

Following chapters you might have to read, to use the Menu:

- You can get an overview of the features in: [The features of the Menu](#).
- In the chapters [Getting Started](#) (look at the Index window) the essential functions are described.
- The chapters [Bugs, Limits, Help](#) (look at the Index window) may protect you for unwelcome surprises.
- The Chapter [Avoid Conflicts with other Programmes](#) may give you hints exactly to do that.
- you have to know the [Edit keys](#) for customising your menu.
- In the chapters [How to do](#) (look at the Index window) the important Actions are described step by step.



## **Thanks to...**

The first Thanks go to John M. Merrill - the first registered User.

Also my beta-testers Michael Kreitmann and Frank Heil have my thanks.

Also a praise to Schulman et al. Without "Undocumented Windows" I could not write the Menu.

## Insert a New Menuitem

### In standard, non-sticked Menus:

#### Comment

- At first you have to popup the Menu with the right (or middle) Mousebutton.
- After that, you have to select (highlight) a Menuitem with the left Mousebutton to reference the point of insert.
- At the same time, push the **Ins-Key**.
- and release the Mousebutton.
  
- In the following dialogue you can declare the characteristics of the new Menuitem.

### In sticked Menus:

#### Comment

#### *Either:*

- You can highlight the Menuitem.
- After released the Mousebutton, push the **Ins-Key**.

#### *Or:*

- Push over the Menuitem to insert the right (or middle) Mousebutton, and select in the Editmenu **Insert Item**.
- In the following dialogue you can declare the characteristics of the new Menuitem.

#### *Or:*

Use the Drag'n Drop Method with the File-Manager.

## Add a New Menuitem

There are the same methods, described in [Insert a New Menuitem](#), with following differences:

- You have to press the **F7**-key instead of the Ins-key.
- The new Menuitem will be add **after** the current highlighted Menuitem.

## Edit an Existing Menuitem

There are the same methods, described in [Insert a New Menuitem](#), with following differences:

- You have to press the **F6**-key instead of the Ins-key.

## Delete an Existing Menuitem

### In standard, non-sticked Menus:

Comment

Comment to delete a Dividing Line

- At first you have to popup the Menu with the right (or middle) Mousebutton.
- After that, you have to select (highlight) the Menuitem to delete.
- At the same time, push the **Del-Key**.
- and release the Mousebutton.
- Confirm following Dialogue.

### In sticked Menus:

Comment

*Either:*

- You can highlight the Menuitem
- After released the Mousebutton, push the **Del-Key**.

*Or:*

- Push the right (or middle) Mousebutton over the Menuitem to delete,
- and select in the Editmenu **Delete Item**.
- Confirm following Dialogue.

### Comments

- If you can't remember all the Edit key, you also can choose the actions via **F5** and the dialogue Configuration.
- Naturally you can't edit Menuitems in Drive Directory.
- If you want to insert a new Menuitem in an empty Submenu, highlight the Menuitem, which contains the empty Submenu and press **Ins**.
- If you insert, add, change or delete a Menuitem in a Program-Group, the changes will also effect the Program-Manager.

### **Comments**

- Searching Directories and Drive Directories can't be edit in sticked Menus.

### **Comment to Delete a Dividing Line**

It is a little bit difficult to select a **horizontal Dividing Line**, but principle possible. The selection of a horizontal Dividing Line has no visual feedback by highlighted the Menuitem.

To delete a **vertical Dividing Line**, you have to select the continuation of the Head of the Menu in the second column.



## Drag'n Drop: Open any File using the File-Manager and a Sticked Menu

Often I want to view the contents of a file, that is not a document. If the file is not too huge, I start the notepad and use File - Open. If the file is bigger than 32 KB I have to use another Editor. This seems to me to be too complicated. Why not use a simple Drag and Drop with the File Manager, to open any file with any Editor?. You can create a Submenu (maybe called 'Browsers') and install all your preferred Text- Hex- and docs-viewer in this Submenu. You can use a Sticked (Sub-)Menu as Drag'n Drop panel. If you drag files from the File Manager and drop them over a MenuItem, the files are sent as arguments to the program installed as MenuItem. Of course, You can also install your Zip- and Unzip batchfiles in one of these Submenus.

- Open the Submenu, which contains a MenuItem with a file viewer (f.e. NOTEPAD.EXE) and stick it on the desktop by using the pin.
- It is recommended to click a second time on the pin, so it is completely Sticked. The Sticked Menu is now every time in the front of the desktop.
- Open the File-Manager.
- Choose the file to open in the File-Manager and
- Drag the file with the left Mousebutton over the MenuItem, which contains the file viewer and drop the file.
- Now the program starts and normally opens the file.

Of course the file viewer must handle the file. Try to open a text file with Paintbrush will fail. Some programmes may also handle more than one File to open.

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 sent as one package to the application.

## Drag'n Drop:

### Insert Files as Menuitems in a sticked (Sub-)menu

A very comfortable way to install new programmes or documents in a Submenu is using drag and drop with the File-Manager.

- Open the Submenu, which should contain the new Menuitem and stick it on the desktop by using the pin.
- It is recommended to click a second time on the pin, so it is complete sticked. The sticked Menu is now every time in the front of the desktop.
- Open the File-Manager.
- Choose in the File-Manager the file(s) to install and
- drag the file(s) with the left Mousebutton over the menuitem, on which position the program or document should insert.
- Press the **Ctrl**-key to add the new Menuitem after the position.

*or*

- Press the **Alt**-key to insert the new Menuitem ahead of the position.

*and*

- drop the file(s).

## The Program Manager as Menu

### Comment

It is possible to include the whole Program Manager as a Submenu containing the Program-Groups as further Submenus with the Programmes as Menuitems. The Icons are supported, but not the short-cuts and the option starting Programmes icon sized.

- To include the Program Manager in a Submenu as a Menuitem, the same steps are necessary, described in Insert a Menuitem respectively in Add a Menuitem.
- Select in the Edit dialogue the option **Progr.-Manager**. The suggested declarations for **Filename** must, and for **Label** can be accepted.

## A single Program-Group as Menu

### Comment

It is also possible to include a single Program-Group (a sub-window of the Program-Manager) as a Submenu in the Menu.

- To include a Program-Group in a Submenu as a MenuItem, the same steps are necessary, described in Insert a MenuItem respectively in Add a MenuItem.
- Select in the Edit dialogue the option **Program Group**.
- Use **Browse** to pick an existing Group file (\*.grp) in your Windows-directory.  
(If you pick Group file not installed in the Program-Manager the Menu may be crashed!)

**Comment**

I hope the Program-Manager and the Program-Groups are quite transparent implemented, but there are still some problems.

However, you sure recognised, the Icons are considerably build slower than in normal Submenus.

I read in journal, that a serious replacement for the Program-Manager 'must' support the group-files of the Program-Managers. OK, I've done it, although it doesn't make much sense to me.

## Edit Menuitems of a Program-Group

### Comment

I hope, it is designed transparent to you.

You can use the normal methods to edit the Menuitems in a **Program-Group** (also **Program-Manager**) build as Submenus.

You have to notice, that all changes (insert, add, edit, delete) will have also an effect to the Windows Program Manager!! For example: If you delete a Menuitem in a Program-Group, the installation of the corresponding Program in the Windows Program Manager will also be deleted.

But there are following differences:

- In contrast to normal Menuitems it doesn't make a difference using **Add** or **Insert**. At all: the positions of the menuitems are taken over in an accidental order.
- Of course you can only insert normal Menuitems (no Submenus or else) in a Program-Group. In the Edit dialogue this limitation is presented with the option **PM-Item**.

### **Comments**

- The Icons are supported, but the short-cuts and the option starting Programmes icon sized will have no effects.

## Menus with a 3-D-Effect

### Comment

Instead of the normal appearance of Windows Menus you can supply the menus with a 3-D-Effect (At first it was only a fashion, it becomes more and more a standard. The programmers do anything for the users!).

Then you can also customise the Font, using by the Menu.

- Select in the dialogue General Properties the option **Graphic**.

If You like the Look of the Desktop Menu of OpenLook, you can change the appearance of the Menus.

Unfortunately this option is not changeable in a dialogue.

Use the editor Notepad, open the file RRKMENU.INI and change in the section [General Properties] the entry OWL\_LIKE=0 to OWL\_LIKE=1. Before restarting the Menu, it is recommended to delete all the files RRKIDX.\* and RRKDAT.\*, because the Icons are different processed with this look.



## Choosing the Menu-Font

### Comment

To change the Menu font, it is necessary to check the Option Graphic.

- Select in the dialogue General Properties the button **Fonts** and choose in the Windows standard dialogue your preferred font.

## Use Icons in the Menu

Comment

Comment 2

One of the highlights of the Menu is the feature, showing Programmes and Documents automatically by their Icons. For Programmes, which haven't an Icon, like DOS-applications, default Icons are used. The first call needs a little bit more time to extract the Icon and write it in the database (RRKDAT.xxx and RRKIDX.xxx in the directory of RRKMENU.EXE). But normally the appearance is quite fast. If you have a very slow computer you may deactivate this option.

- Select in the dialogue General Properties the option **Icons**.
- In the field **Icon Size** you can choose the preferred icon size. 32 pixels is the normal size, also showed by the Program Manager. 24 and 16 pixels are equivalent smaller.

**Comment**

If you change the Font- or the Icon size, while sticked Menus exist on the desktop, you can refresh this sticked Menus with the **F12**-key. (This may be fixed in a later version.)

## Starting Options

### As Shell:

Normally Windows starts at first the Program Manager for Windows Shell.

If you like the Menu, and find the Program Manager unnecessary for you, you can use the Menu as Windows Shell:

(Comment: At my experience the menu is more economically with the System resources as the Program Manager.)

- Select in the dialogue General Properties the option **As Shell** and restart Windows.

You can enjoy the purity starting Windows with a complete empty desktop (All wishes to order and purity tend to negate all things to nothing), and like a deus ex machina the Menu waits for you in the background.

But the Menu also supports a Submenu, called **Autostart**. It functions in the same way as in the Program Manager of Windows. Create this Submenu, if you want start applications at the beginning of Windows. The Menu also supports the **run** and **load** directives in the WIN.INI.

### Save the Desktop by ending the Menu

If you have collected our preferences Applications and our Project-Files (may realised with **Searching Directories**.) in sticked Submenus on the desktop, you can induce the Menu to restore these sticked Submenus by restarting the Menu.

Submenus, build by **Drive Directories**, will not be restored.

- Select in the dialogue General Properties the option **Save DT**.

## Using the middle Mousebutton

Although the Microsoft Mouse has only 2 mousebuttons, 3 mousebuttons are supported by Windows. (It depends also by the used mouse driver)

If you have a mouse with 3 mousebuttons, it is recommended using the 3th, middle mousebutton to popup the Menu.

- Unfortunately you can't test the 3th mousebutton in the Windows Control. For precaution at first it is recommended to unhide the Menu icon (the black arrow) by using **Hide Icon** in the Submenu **Menu Setup** or stick the Submenu **Menu Setup** on the desktop.
- Select in the dialogue General Properties the option **middle Mousebutton**.
- Test the middle mousebutton.
  
- If your Mouse doesn't support the 3th mousebutton, undo this option.

Look also: [Avoid Conflicts with other Programmes](#)

## Improve the Speed of the Menu

If you haven't serious problems with our RAM (286 / 1 MB) you can improve the speed of the Menu by

- selecting in the dialogue General Properties the option **Index in RAM**.

Without this option the Menu needs permanently only ~25 KB non-discardable RAM, with this option ~100 KB more (It depends by the number of Document-Associations and registered Icons in the Menu-Database).

Commend

## Avoid Conflicts with other Programmes

The Menu is a kind of TSR, hooking the mouse actions.

Unimportant which application is actual active, pressing the right (middle) Mousebutton the Menu comes up.

This may be causes problems.

To minimise these problems, the Menu is very jealous of other Programmes using the right (middle) mousebuttons. More and more Programmes supports the right mousebutton for local edit options. In this case, you normally can't access this features of the Programmes, because every time the Menu is popued.

- If you have a mouse with 3 mousebuttons, it is simple to solve this problem: Use the middle Mousebutton for the Menu!
- Otherwise can use the Option **Only Desktop** in the Submenu **Menu Setup**. Then the Menu only will popued, when the mousecursor is over the background of the desktop. In the applications the right mousebutton will function in the normal way.  
If you select the option **Alt + Pause PopUp** in the dialogue General Properties, You can still popup the menu independent of the position of the mousecursor by pressing the Alt- and Pause- Key.  
If you set the entry **KeySub** in the section [General Properties] with the value 2 or 3, the Menu will only popued, if the cursor is over an non-client-region (f.e. the Titelbar of a window). In the client-regions of a window the local Popupmenu of the application then works in a normal way.

**Comment**

This option is used by default.



## Hide the Program Icon

One of the aim of the Menu is, to clean up the overcrowded Desktop. By consequence you can also hide the program icon of the Menu (a black arrow).

*Either:*

- Select in the System menu of the program icon the menuitem **hide Icon**.

*or:*

- Select in the Submenu **Menu Setup** the Menuitem **hide Icon**.

To unhide the Icon:

- Of course you can unhide the icon only with the Menuitem in the Submenu **Menu Setup**.



visible.



hidden

## only Desktop

If you want, that the Menu will only popuped, when the mousecursor is over the background of the Desktop, you can select **only Desktop** in the Submenu **Menu Setup**.



1. The Menu will popuped on every place of the desktop.



2. The Menu will only popu

ped, if the mousecursor is over the background of the desktop.



3. The Menu will only popu

ped, if the mousecursor is over the background of the desktop **or** if the mousecursor is over a title bar of a window

You can togle between 2 and 3 with pressing **Ctrl** while selecting **only Desktop**.

Look also: [Avoid Conflicts with other Programmes](#)

## Using Hotkeys

To launch Programmes or execute Internal Commands with the keyboard, you can use Hotkeys. In contrast to ShortCuts for Hotkey it is not necessary, that a Menu is popued.

If you want use Hotkeys, the option **Alt+Pause-Popup (HotKeys)** in the dialogue General Properties must be activated. It is also recommended that the option **Index in RAM** is activated, otherwise the reaction of the keyboard may be slower.

You can define Hotkeys in two ways:

### 1. In the Edit-Dialogue:

If you define a new MenuItem or edit an existent MenuItem, you can also define a Hotkey in the editbox **Hotkey** to execute the command line with a key combination. If the editbox Hotkey has the focus, push the key combination.

If for command line (also the parameter, working directory, window size and the Icon is relevant) a Hotkey is defined, the key combination is showed in the editbox Hotkey. If you declare a new Hotkey, and the key combination is already defined for another command line, the old association will be overwrite.

### 2. In the Dialogue Hotkeys:

If you want to have an overview of the defined Hotkeys, you can use the Dialogue Hotkeys to define and undefine Hotkeys.

## Using Context-Sensitive Menus

Context-Sensitive Menus extend the functionality or the comfort of access of a defined Window-Class like EditBoxes, ListBoxes or the DOS-Box. You can get the associated Context-Sensitive Menu with the right mousebutton, if the mouse cursor is over a defined Window-Class.

All defined Context-Sensitive Menus are described in the file **ADDONS.WRI**.

If you want you the Context-Sensitive Menus, the option **Context-Sensitive Menus** in the dialogue General Properties must be activated.



## Using the Menu without a Mouse.

If you want to use the Menu on a Computer, which haven't a mouse, it is principle possible.

- You can use the **Alt - Pause**-keys to popup the Menu.
- Use the **Esc**-key (maybe several times) to close the Menu. (excepting the sticked Submenu)
- To launch a program use the **Return**-key.
- Use the **Up** and **Down Arrows** on the Keyboard to highlight a Menuitem.
- Use the **Right Arrow** or the **Return**-key on the keyboard to open a highlighted Submenu.
- Use the **Left Arrow** or the **Esc**-key to close the last Submenu.
- To stick a Menu use the **Return**-key, when the Menu head is marked (look at the pin).

For the next actions a sticked Menu must have the focus (showed by two thin lines in the Menu head).

- Use the **Shift-PgnUp**-keys to roll up the sticked Menu.
- Use the **Shift-PgnDown**-keys to roll down the sticked Menu.
- Use the **Shift-Esc**-keys to hold a sticked Menu on the top of the desktop.
- Use one or two times the **Shift-Esc**-keys to close a sticked Menu.
- Use the **F9**-key to arrange all sticked Menus on the left side of the desktop.
- Use the **F11**-key to arrange all sticked Menus on the top of the desktop.
- Use the **Shift** and the **Arrow**- keys to move a sticked Menu

## Install the Menu in a Net environment

If you want to install the Menu on a File-Server, you can declare the place of the RRKMENU.INI as the argument of RRKMENU.EXE

Supposing the File-Server is associated with `d:\` and the Menu is installed at `d:\windows\rrkmenu` and the local users harddisk with the users INI-files is associated with `c:\`, so f.e. you can install the Menu in the Program Manager with the commandline `d:\windows\rrkmenu\rrkmenu.exe c:\windows\rrkmenu`

In this directory `c:\windows\rrkmenu` the Menu will also create the Files `RRKDAT.*` and `RRKIDX.*` for the Icons database.

## Declare Icons for Non-Windows Programmes

DOS-Programmes, PIF, COM and BAT-Files don't have individual Icons. These Programmes are showed with standard icons.

But there are two ways to declare individual icons:

The first way is to declare an Icon in the edit dialogue.

- Highlight the menuitem containing the program.
- push **F6**,
- and choose via Icons a icon-file.

This way has the disadvantage, that the Icon is only declared for the current icon-size and colour-resolution. The information about the new icon will also be deleted, if you change the Style from or to **Standard** in the general Properties dialogue.

Another way to declare a new icon for a DOS-application is to create an icon with an Iconeditor. Store the icon with the same name of the application in the path of the application. F.e. if you want to declare an new Icon for `c:\progrpath\dosapplic.exe` create an Icon `c:\progrpath\dosapplic.ico`. If you don't see an effect in the menu, you have to delete the files `RRKDAT.*` and `RRKIDX.*` in the directory of the menu.

(Comment: Don't use both methods parallel, because if you delete the files `RRKDAT.*` and `RRKIDX.*` all Icons, you declared with the first method, are discarded.

I know, the two methods aren't very outperformed. Maybe it becomes better in the next releases.)



## Using ShortCuts

You can assign Hotkeys for your Submenus.

If you push a combination of **Ctrl** and a char, while the menu is popuped up, the declared Submenu is popuped up as the main - Popupmenu.

- Call in the popuped Menu with **F5** the Configuration dialogue.
- Push the button **ShortCuts**: The dialogue ShortCuts will come up.
- In the left list all declare Hotkeys and in the right list all Submenus are listed.
- Write a char below **Ctrl** - and select a Submenu, (f.e. T)
- Push the button **Associate** (f.e. Tools).

If you no push `Ctrl - t` in the popuped Menu the Submenu `Tools` will appears.

## Use the right Mousebutton with a Second (Sub-)Menu

If you normally use the **middle** mousebutton to popup the Menu, you can declare a second (sub-)Menu for your **right** mousebutton. This second Menu you can popup up with your right mousebutton over the background of the desktop or over a title bar of a window.

- Use Notepad to open the file RRKMENU.INI.
- Look in your INI-file for the name of the Submenu, you will popup with the right mousebutton.
- Write this name (f.e. My Submenu) behind the entry `SpezialRightMouseButton=` in the section `[General Properties]:`  
(f.e.: `SpezialRightMouseButton=My Submenu`).

## Change the style of the Menu

In the dialogue General Properties you can choose one of three graphical styles for the Menu.

**Standard** is a simple style, but it has the advantage the menuitems are small and the Submenus are build quickly.

The style **OpenLook** is overtaken from a Unix X-Windows GUI.

The style, called **Beta** is just an one-evening-production.

If you change the style to or from **Standard** all icon database files RRKIDX.\* and RRKDAT.\* are deleted.

## Using a Submenu as a Desktop-Panel



There is the possibility to install the important programmes in one or more submenus and stick these Submenus on your desktop.

(Komment: In [Drag'n Drop: Drag'n Drop: ... in Sticked Menus](#) you can read how to use sticked Menus.)

If you mean the description of the menuitems are not needed, you can let the Menu to show Submenus without the text.

To do this, there are two possibilities:

### 1.

- While sticking a Submenu on the desktop press **Strg+Shift**.

### 2.

The first method has the disadvantage, that the pinned Menus not automatically restored without the description-text. (See [Starting options](#))

If you want a Submenu allways showed only with the icon you have to:

- open the `RRKMENU.INI` with Notepad.
- and look for the section `[My Submenu]`, which shall showed without the description-text and insert the line  
`0=1`  
below.

## Using Drag'n Drop-Panels

Because the title bars of Drag'n Drop-Panels are to small to show the pin and the Roll-Up-Button, you have to tongle the aktive part of the title bar with the **Tabulator**-key

At first the panel looks like:



In this state you can roll up and down and move the panel with the mouse.

After pressing **Tab**:



Now you can only move the panel with the mouse.

One time more pressing the **Tab** you can see following picture:



In this state you can click one (to hold the sticked Menu on the top) ore two (to close the sticked Menu)

times the pin.

If you push a third one the **Tab** you reach the first state again.

Of course also the keys described in Using the Menu without a Mouse in the normal way.

## Running a Windows incompatible DOS Program

Some DOS programmes you can't run with Windows, because they use their own DOS-Extender or the 32-bit Modus of the 386 (like Autocad, gcc or some Databases), or they need a direct access to the Hardware (like some Disk-Utilities or many games).

Now, the Menu really **doesn't** have a genius windows-driver, to solve all this problem, but anyway, you can start such programmes with a menuitem.

You can declare the DOS-Program in the **command line** of the Edit dialogue in this way:

```
MNU_WIN_REBOOT c:\command.com /c c:\my_dir\my_prog my parameters
```

`c:\command.com` ist der Standard-command interpreter of DOS.

If necessary change the location and the name of your DOS command interpreter. The flag `/c` tells `command.com` of MS DOS to execute only the command `c:\my_dir\my_prog my parameters` and exit after that itself.

(Look in the documentation of your `command.com` for an equal switch)

Schauen Sie in der Dokumentation ihres Befehlsinterpreter nach dem entsprechenden Parameter). At least `c:\my_dir\my_prog my parameters` is the program to execute.

The **Label** of this Menuitem is ad libitum.

You can also declare an individual Icon for this menuitem, but not - naturally - a windows-size.

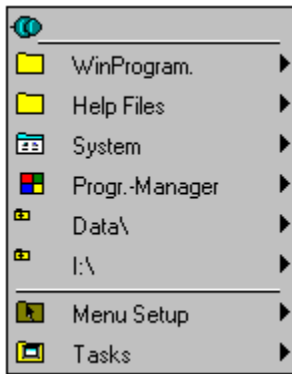
If you now select this Menuitem, Windows will shut down and the program `c:\my_dir\my_prog` will execute. If `my_prog` ends Windows will automatically restart.

If you declare in the command line only `MNU_WIN_REBOOT`, Windows will immediately restart after shooting down.

If you press the **Shift**-key selecting this Menuitem the system (computer) will be rebooted.

## An Example Menu

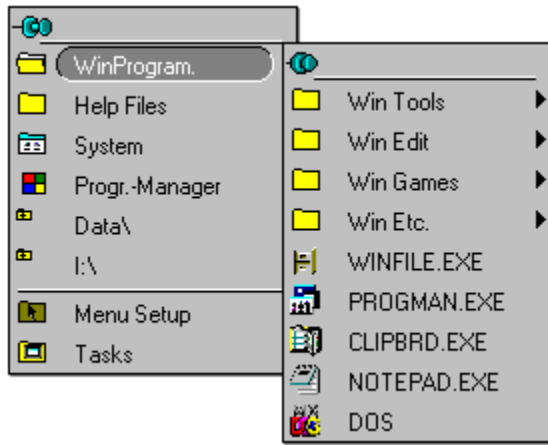
(Hint: Click on the Menuitems.)



## An Example Menu

(Hint: Click on the Menuitems.)

### A Submenu





## A Submenu

If you click on a MenuItem containing a Submenu, the Submenu will be open.

You can group our applications and documents in several Submenus, similar to the Group Windows of the Program Manager.

A Submenu can contain all other types of Menuitems, including more (Sub)Submenus.

A Submenu is similar to a Subdirectory of the harddisk with one difference:

Submenus with the same name in the hierarchy of the Menu have only two references to **one** Submenu.

Submenus have more likeness to links, known under UNIX systems, than to a DOS - path. This have f.e. the consequence, that a Submenu named `My Submenu` can contain recursive the Submenu `My`

`Submenu`.

Every Submenu (excepting the Submenu **Tasks**) can be sticked with the green pin on the desktop.

**Submenu**

More SubSubmenu

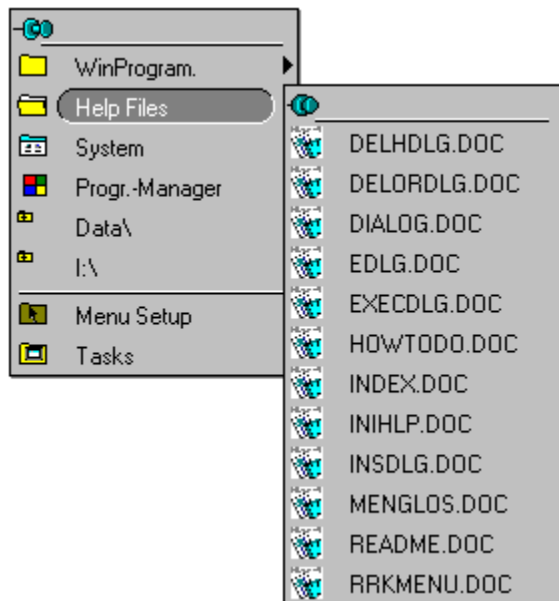
**standard MenuItem**

A standard MenuItem contains a Program or a Document to launch.

## An Example Menu

(Hint: Click on the Menuitems.)

### A Searching Directory



## **A Searching Directory**

All files are listed matching to a DOS-ambiguous file name (here f.e. `c:\d\c\menu\*.doc`)

## **A Searching Directory**

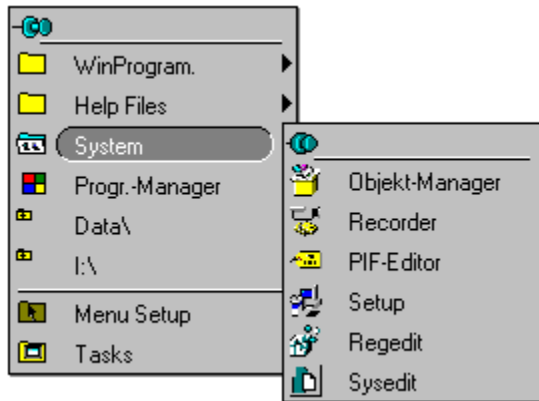
Searching Directories are useful to hold all document-files of your actual project in one Submenu, so you have a quick access to your often used files.

F.e. I create a Submenu named `Help files`, and insert only one Menuitem in this submenu. This Menuitem is a searching directory with the filename `c:\d\c\menu\*.doc`. In the Submenu `Help files` all my help-files are listed. I sticked this Submenu on the desktop, and arrange it on the top of the desktop. Now the circumstance of the use `File / open` in Winword is obsolete. I have only to roll down the sticked menu, to choose a file to open in Winword, and to roll the menu up.

## An Example Menu

(Hint: Click on the Menuitems.)

### A **Program Managers Group**



### **The installed Programmes of a Program Group**

All installed Programmes of a Program Group are listed with their icons.



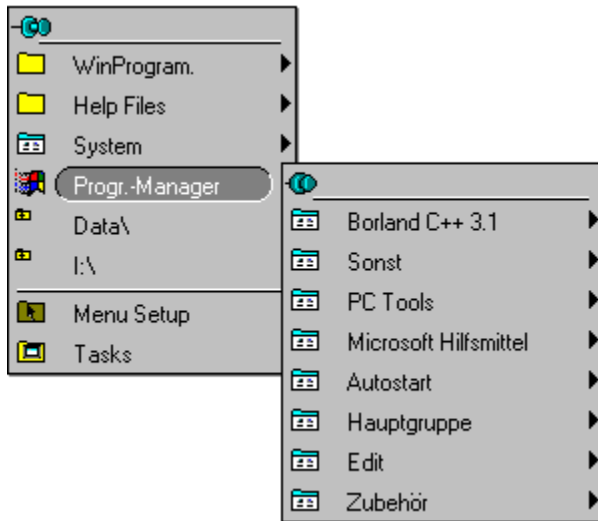
### **A Program Group**

A Program Group takes over its Items from a Program Managers Group file (\*.grp). With Program Groups you can extract a single Subwindow of the Program Manager and include it in your Menu.

## An Example Menu

(Hint: Click on the Menuitems.)

The Program Manager



### **The content of the Program Manager**

In this Submenu all Subwindows of the Windows Program Manager are listed as Submenus.

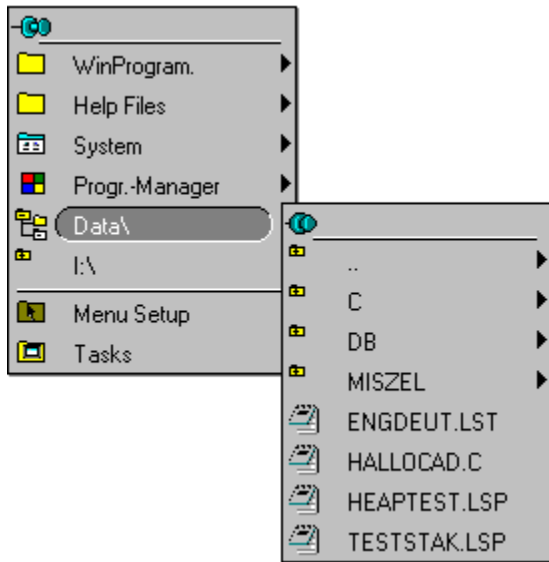
### **The Program Manager**

With the Menuitem **Progr. Manager** you can insert the whole Program Manager in your Menu. All Program Groups (Subwindows) of the Program Manager are listed as Submenus.

## An Example Menu

(Hint: Click on the Menuitems.)

### A Drive Directory 1



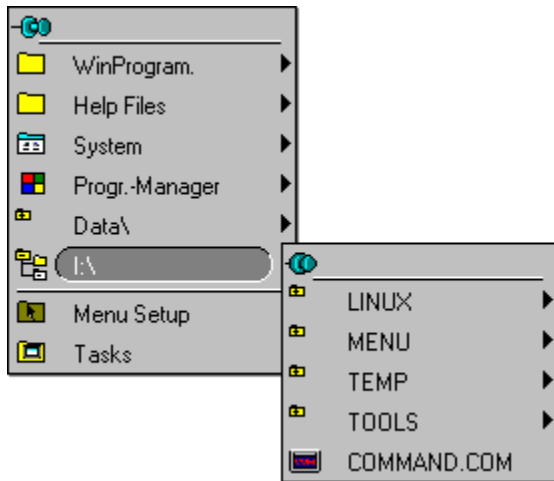
**Drive Directory**

All founded Subdirectory of current Subdirectory are listed as further Submenus. Similar to the File Manager, you can open this Submenus to look the contents of the Subdirectory.

## An Example Menu

(Hint: Click on the Menuitems.)

### A Drive Directory 2



**Drive Directory**

All programmes and documents of a subdirectory are listed as Menuitems.



### **Drive Directory**

You can automatically present the contents of a path of your harddisk in a structure of Submenus. The beginning of a Drive Directory can be a path like `c:\letters\new` or the beginning of an harddisk like `c:\`.

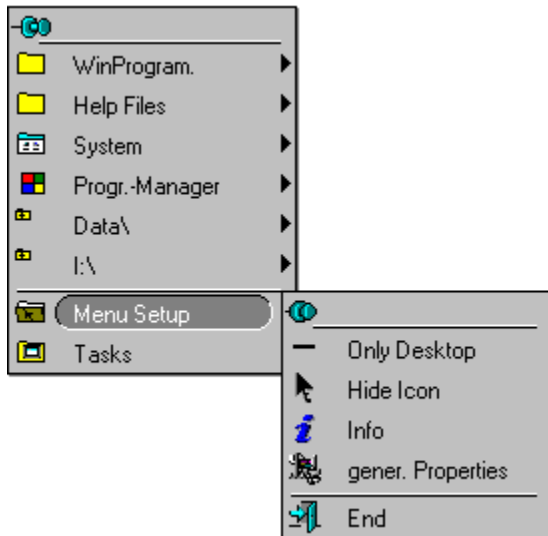
In this Submenus at first all directories as further Subdirectories and second all programmes and documents as Menuitems are listed.

Every time, the Submenu gets too long fitting on your Desktop, a Submenu called more... will be append at the end of the Submenu. In this Submenu the rest of the founded Subdirectories and files are listed.

## An Example Menu

(Hint: Click on the Menuitems.)

The Menu - **Menu Setup**



## **Menu - Menu Setup**

This Submenu is reserved for functions of the Menu. Click on the Menuitems to get more information.

### Only Desktop

If you want, that the Menu will only popuped, when the mousecursor is over the background of the Desktop, you can select **only Desktop** in the Submenu **Menu Setup**.



1. The Menu will popuped on every place of the desktop.



2. The Menu will only popu



ped, if the mousecursor is over the background of the desktop **or** if the mousecursor is over a title bar of a window

You can tonge between 2 and 3 with pressing **Ctrl** while selecting **only Desktop**.

See also: [Avoid Conflicts with other Programmes](#).

## Hide Icon



Visible.



Invisible.

One of the aim of the Menu is to clean up the overcrowded Desktop. By consequence you can also hide the program icon of the Menu (a black arrow).

Normally the program-icon is showed at the bottom of the screen. If don't want it on your desktop, hide it.

**Info**

Shows the program information.

**gener. Properties**

Opens the dialogue general Properties to change several customisable features of the Menu.

## **End**

Ends the Menu.

If the Menu is running as Windows Shell or if you pressing the **Ctrl**-key while selecting End, after confirming Windows will shoot down.

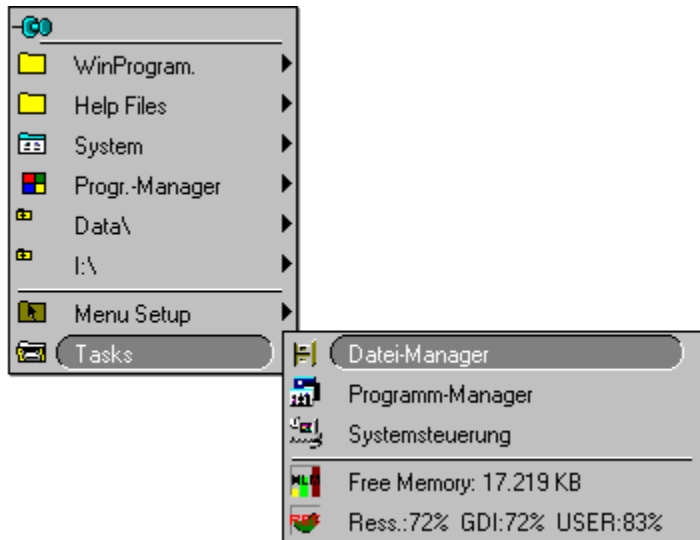
See also: [Running a Windows incompatible DOS Program](#)



## An Example Menu

(Hint: Click on the Menuitems.)

### The Menu Tasks



### **The Menu Tasks**

In this Submenu - the only one without a pin to stick the menu on the desktop - you can switch between the programmes running actually at Windows. (Click one one Task for more information.)

If you use the Menu as Windows Shell, a menuitem **Save Wins DT** is added.

With this menuitem you can save all positions of the applications (windows) on the desktop. If you restart Windows, all Application will be restored exactly in the same positions (exception of the applications are in [Autostart]). If you push **Shift**-key while selecting **Save Wins DT** the menu will not restore the windows at the following starts of Windows.

This option has only one disadvantage. Of course (!) the Menu can't figure out, which documents are loaded in the applications, the document-based applications will be restored empty.

### **The Tasks**

All visible windows (Programmes) are listed in the order of their appearance. If you press the **Shift**-key while opening the Submenu **Tasks**, also the invisible windows will be listed. By selecting a Menuitem the window will be brought to the top of the desktop. But if you press the **Ctrl**-key while selecting a Task, the corresponding Program will be ended, respectively the corresponding window will be closed.

**Free RAM**

Shows the free RAM of Windows.

This menuitem has no further function.

### **Free Resources**

Shows the free Windows-system-resources of the graphic engine (GDI) and for the windows-organisation (USER). The System resources, also showed by the Program Manager under Info, is the less value of both.

This menuitem has no further function.

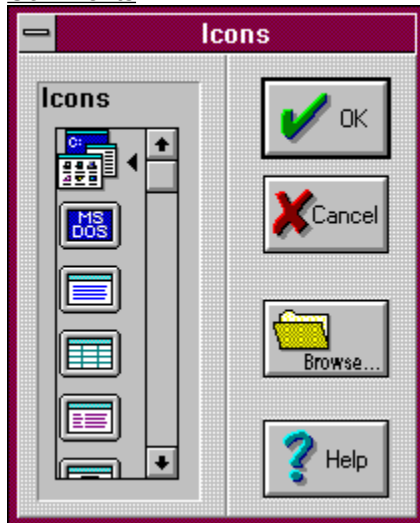
## Der Dialog - Select Icon

(Hint: Click on the buttons and fields for more Information.)

How to get the Dialogue

Explanations

Comments



## How to get the Dialogue

Push in the dialogue Menu - Edit the button **Icon...**

### **Explanations**

After choosing a Iconfile in the file-dialogue, all Icons of the file are listed. After selecting one of these Icons the program, declared in the Editdialogue will showed with this Icon.

If you have declared a document in the Editdialogue, the change of the Icon will not have any effect in the Menu.



### **Comments**

If you change the size of the icons showed in the menu or if you change the number of showed colours under Windows, the declared Icons for programmes showed in the Menu are discarded.

Another - independent of the iconsize and the colour-resolution - way to show non-Windows with an Icon is described in [Declare Icons for Non-Windows Programmes](#).

**Icons**

All Icons of the selected Iconfile are listed. The black triangle shows to the selected Icon.

**OK**

The new Icon for the program is only overlooked, when you confirm the Editdialogue

**Cancel**

Close the dialogue without defining a new Icon

**Browse**

Opens a Standard-Dialogue to choose an Iconfile (\*.ICO;\*.EXE;\*.DLL).

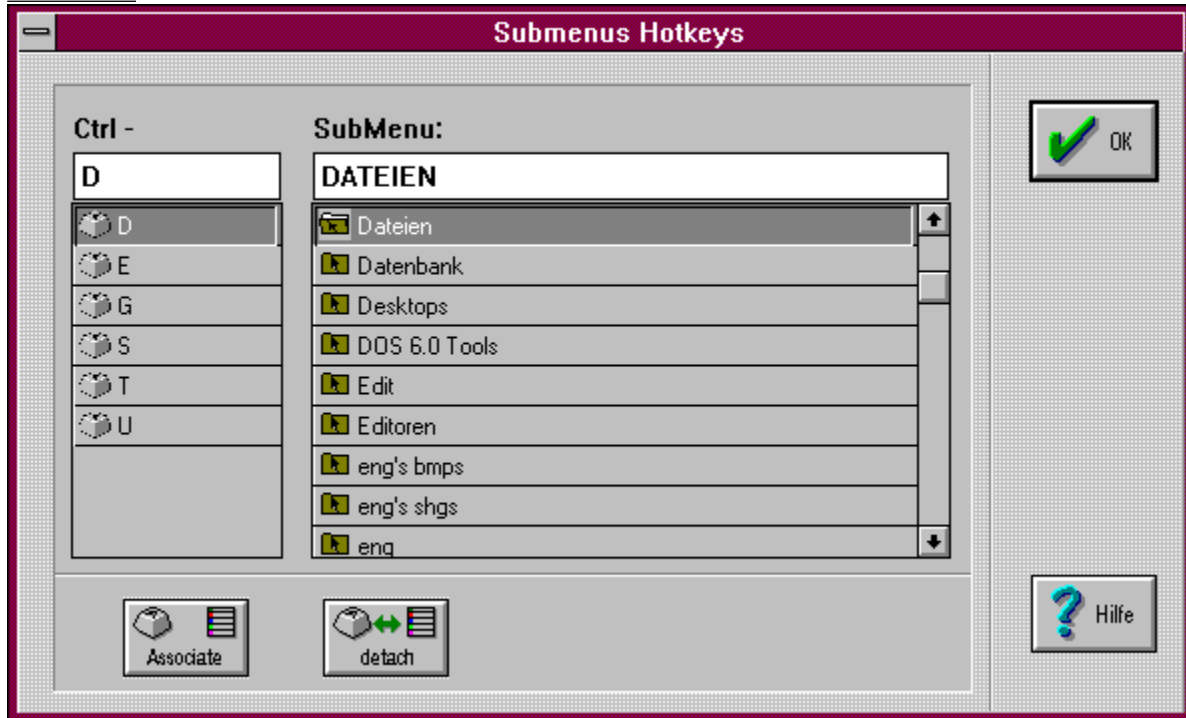
## The Dialogue - ShortCuts for Submenus

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



### How to get the Dialogue

Select in the dialogue Configuration the button **ShorCuts**.

**Explanations**

If you popped the Menu you can jump to a Submenu with a Hotkey.  
These Hotkeys you can declare in this dialogue.

See also: [Using ShortCuts](#).



**Comments**

It is also possible to declare the right Mousebutton with a second Submenu.

## **Ctrl - Hotkeys**

All Hotkeys declared for the menu are listed. Click on an item to look the assoziated Submenu

## **Submenus**

All Submenus, defined in RRKMENU.INI, are listed.

### **Assoziare**

If you write a char below **Ctrl-** as the hotkey and you have selected a Submenu you can associate the Hotkey with the Submenu.

**Detach**

Select a Hotkey-char (the assoziated Submenu will automaticly be selected) and push **Detach**, to detach the Assoziation.

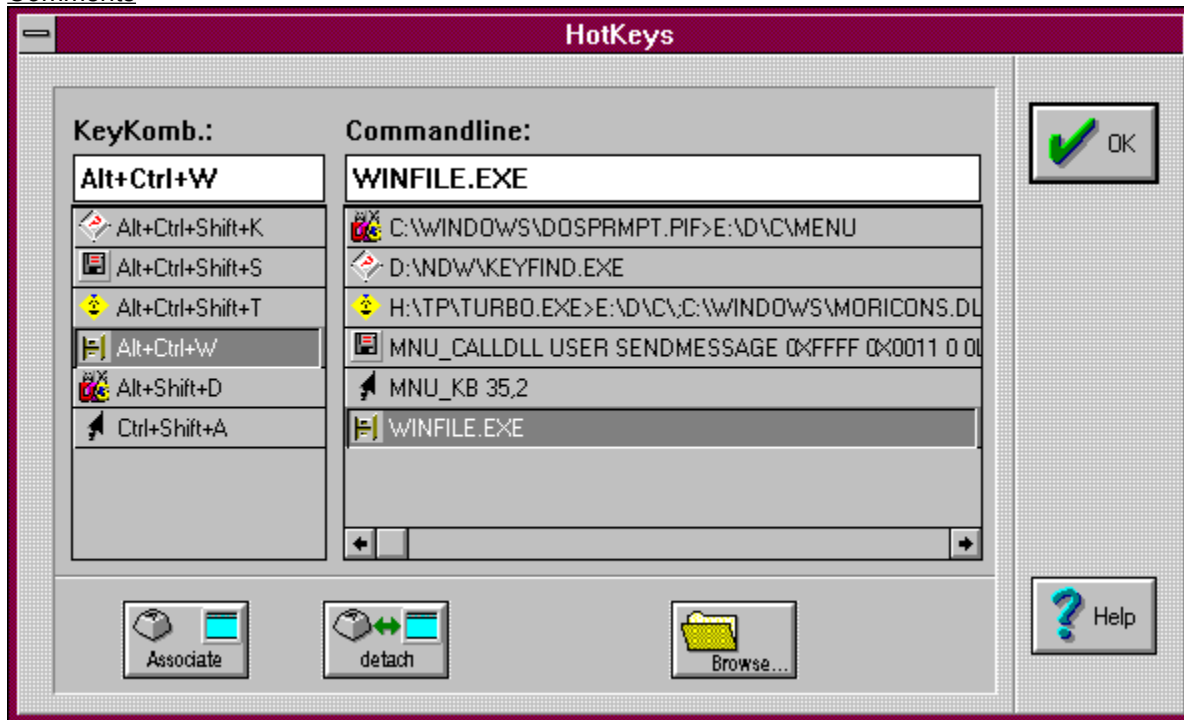
## The Dialogue Hotkeys

(Hint: Click on the buttons and fields for more Information.)

[How to get the Dialogue](#)

[Explanations](#)

[Comments](#)



### How to get the Dialogue

Select in the dialogue Configuration the button **Hotkeys**.

### **Explanations**

In this dialogue all key combinations for Hotkeys, which are associated with a command line, are listed. It is also possible to declare Hotkeys for a MenuItem in the Edit - dialogue.

In the field **Commandline** are all command lines - inclusive the parameters, the working directory and window-size valid. You can also declare an Internal Command as a command line.

A new key combination for a Hotkey you can define, if you put the cursor in the field **KeyComb.:** and press the keys.



### **Comments**

It is also possible to declare Hotkeys for a MenuItem in the Edit - dialogue. But it may be better to have an overlook to all declared Hotkeys, because the command line - Hotkey - association may be ambiguous.

**KeyCombination**

All defined Hotkeys are listed.

**Command Line**

All command lines, which are associated with a Hotkey, are listed.

## **Associate**

If You have declared a **key combination** for a new Hotkey and a command line (maybe via **Browse**) you can use **Associate** to define a new Hotkey.

**Detach**

If you want to undefine a Hotkey select the **Key Combination** and push **Detach**.

## Hotkey

With a Hotkey - a key combination with the Alt and/or Ctrl- and the Shift-key with a char - you can launch Programmes or execute Internal Command with the keyboard every time.  
For a valid Hotkey the Alt- or the Ctrl-key and a char must be defined.

Examples for valid Hotkeys key combinations:

Ctrl+D; Alt+1; Alt+Shift+T; Ctrl+Shift+Z; Alt+Ctrl+Shift+O

**Not valid** are following combinations:

Shift+A; Ctrl+#

The Hotkeys only function if in the dialogue general properties the option **Alt+Pause-Popup (Hotkeys)** is activated.

## Internal Commands

### **MNU\_ADD**

It doesn't make a sense, to use this internal command at an other place, as in the Edit-Menu of sticked Menus.

### **MNU\_ARRANGE\_LEFT**

Minimises all sticked Menu - excepting the Menu only with icons and the sticked Drop-Down Menu - and orders them at the left side of the desktop.

### **MNU\_ARRANGE\_TOP**

Minimises all sticked Menu - excepting the Menu only with icons and the sticked Drop-Down Menu - and orders them on the top of the desktop.

### **MNU\_BACKGROUND**

See Only Background

### **MNU\_CALLDLL**

Calls an extern DLL-function.

See also: Call a DLL-Function

### **MNU\_COMMAND:**

**Doesn't exist!** because you don't need it. If you want to prompt for a command-line, select a menuitem, and press the **Ctrl**-key while release the mousebutton.

In the dialogue Menu - Execute you can declare a commandline.

### **MNU\_CONFIG**

Calls the dialogue Konfiguration.

### **MNU\_DEL**

It doesn't make a sense, to use this internal command at an other place, as in the Edit-Menu of sticked Menus.

### **MNU\_EDIT**

It doesn't make a sense, to use this internal command at an other place, as in the Edit-Menu of sticked Menus.

### **MNU\_END**

Ends the Menu or Windows; with or without confirmation.

If you use the Menu as Windows-Shell, Windows will shoot down after a confirmation. If you press the **Shift**-key while selecting this menuitem, Windows will shoot down without confirmation.

If the Menu is a normal Windows-application, the Menu ends. You can also quit Windows if you press the **Ctrl**-key. If press the **Ctrl**- and the **Shift**-key Windows will also shoot down without confirmation.

### **MNU\_HIDEICON**

(Symbol verstecken = Hide Icon)

See Hide the Program Icon

### **MNU\_INFO**

Calls the info dialogue.

### **MNU\_INS**

It doesn't make a sense, to use this internal command at an other place, as in the Edit-Menu of sticked Menus.

### **MNU\_KB**

Send keyboard actions to the activ window.  
See also: [Generate keyboard actions with a MenuItem](#)

### **MNU\_MTW**

Sends a message to a window.  
See also: [Send Messages to a Window](#)

### **MNU\_PROGMAN**

Generate a Submenu with all Programm-Groups of the Program-Manager.

### **MNU\_SDT**

With this command you can save all positions of the applications (windows) on the desktop. If you restart the Menu (or Windows), all Application will be restored exactly in the same positions (expection of the applications are in [Autostart]). If you push **Shift**-key while selecting **Save Wins DT** the menu will not restore the windows at the following starts of the Menu. If you if hold down the Shift-key, while starting the Menu, the Desktop won't also restored.

This option has only one disadvantage. Of course (?!) the Menu can't figure out, which documents are loaded in the applications, the document-based applications will be restored empty.

At the end of the Submenu **Tasks** is a MenuItem **save Wins DT**, which has the this internal command.

### **MNU\_SETUP**

It is only used to declare for the Submenu [Menu Setup](#) a special Icon.

### **MNU\_TASKS**

Generate a Submenu with all Windows (programmes), also the free memory and System-Resources are showed.

If you press the **Ctrl**-key while opening this submenu also all hidden Windows are showed.

If you click on one of this Tasks-menus, the corosponding programm comes into to the front. If you press the **Strg**-key while selecting a Tasks-menuitem, the corosponding programm will be end.

Normaly the syntax for MNU\_TASKS in a MenuItem is:

```
4=1|Tasks|MNU_TASKS
```

But it is also possible to declare a Submenu in this way:

```
[My Tasks-Submenu]  
1=MNU_TASKS
```

### **MNU\_WIN\_REBOOT**

Shoot down Windows without confirmation and restart windows.

You can also execute a DOS-Programm or DOS-command.

**If you want to execute a DOS-Programm it is imporent to declare a command-interpreter (normally command.com). Otherwise the system halts!**

A command-line can look like:

```
MNU_WIN_REBOOT c:\command.com /c c:\my_dir\myprogram
```

c:\command.com is the command-interpreter of DOS. /c tells the command.com, that it only have to execute one program c:\mein\_dir\meinprog and after that, to terminate itself.

This option is useful to execute a DOS-Program, which can not run in a DOS-Box of Windows.

If you don't declare parameters after MNU\_WIN\_REBOOT, Windows will restart with no action on the DOS-level.

If you press the **Shift**-key, while selecting a **MNU\_WIN\_REBOOT**-MenuItem, your computer will reboot (similar to Ctrl-Alt-Del).



## Simulate keyboard actions with a MenuItem

If you want generate system-wide keyboard actions with a MenuItem, you can use the internal command **MNU\_KB**.

MNU\_KB Char,Schiftstatus secondChar,secondSchiftstatus ...

- Char:  
The key (a char or a VK\_scancode declared in windows.h).
- Schiftstatus:  
Shift-, Alt- and/or Ctrl-key as bit-combination.
  - 0 = no Shift
  - 1 = Shift-key
  - 2 = Ctrl-key
  - 4 = Alt -keyThe Shift-Ctrl-Alt-combinations results out of the sum of the bits. To produce f.e. the key-combination d-Alt-Ctrl, you have to declare `D, 6`.
- The single key-statements are seperated with a whitespace.

## Call a DLL-Function

It is also possible to call a function you have wrote in a DLL or call a system function of the Windows API with a menuitem.

The grammar of a command line of a menuitem:

MNU\_CALLDLL Modulname Functionname Parameter

- **MNU\_CALLDLL** is the internal command to point to function in Pascal-konvention.
- **Modulname** is the name of the DLL, f.e. `my_own.dll` (maybe the path to the DLL is required.)  
The most used moles of the Windows-API are USER, GDI and KERNEL.
- **Functionname** is either the name of the function or its exported ordinal-number.
- **Parameter** is a list of the parameter of the function. (The parameters are overgiven in the Pascal-konvention.)  
For the typ of the Parameters you have consider follow conventions:
  - Integer number are written as normal numbers. Negative numbers get an minus sign (f.e. 123 or -321)  
If you want write an number in hex you have to beginn the number with 0X (null-x): f.e. 0X1A4C)
  - Long Interger (long) ends with a L (1234L or 65L)
  - Floating point numbers are written with a point (2.14 oder 12.0).
  - Bytes (a char) are started with a inverted comma ('c or 'x).
  - char strings are enclosed with cuotation marks ("hallo").

### Warning!!

**Be carfull with the parameters. If you declare the wrong type and numbers of parameters you get an GP-fault!**

*Examples:*

A beeping Menuitem:

```
MNU_CALLDLL USER MESSAGEBEEP 0
```

A Menuitem calls a MessageBox:

```
MNU_CALLDLL USER MESSAGEBOX 0 "And good Beye" "Hello" 4
```

Even undocumented Function of Windows can be called:

A Menuitem to arrange all minized Windows at the bottom of the desktop:

```
MNU_CALDLL USER ARRANGEICONICWINDOWS 0XE8C  
(0XE8C is the HWND of the desktop)
```

Arrange all not iconsized Windows next to each other:

```
MNU_CALLDLL USER TILECHILDWINDOWS 0XE8C
```

Arrange all not iconsized Windows overlapping:

```
MNU_CALLDLL USER CASCADECHILDWINDOWS 0XE8C 0
```

The command:

```
MNU_CALLDLL USER SENDMESSAGE 0xFFFF 0X0011 0 0L  
with 0xFFFF=HWND_BROADCAST and 0X0011=WM_QUERYENDSESSION,  
has the function to ask all win-programmes, if they has some unsaved changes.  
(You can add this command line to the Submenu Menu Setup if find it is useful.)
```

The Label of a Menuitem with MNU\_CALLDLL in the command line is ad libitum, also you can declare an other icon.



## Popup a Menu with an extern Program

It is also possible to popup a Submenu of the Menu out of another program. But it is necessary that the RRKMENU.EXE is already running.

To do this, use the function:

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

of the DLL RRKMNDLL.DLL.

Submenu is the name of the Submenu, which shell appears as the main menu. If Submenu = NULL, the normal MainMenu of RRKMenu appears.

Using C as programming language, you can use following procedure:

```
BOOL PopupRRKMenu(char *Submenu){
    HANDLE hrrkmdll;
    FARPROC callRRKMenu;
    BOOL result;
    static BOOL (FAR PASCAL *CallRRKMenu)(LPSTR Submenu) = 0;
    if (!(hrrkmdll = LoadLibrary("RRKMNDLL.DLL"))){
        return FALSE;
    }
    if (!(CallRRKMenu = GetProcAddress(hrrkmdll, "CallRRKMenu"))){
        FreeLibrary(hrrkmdll);
        return FALSE;
    }
    ergebnis = CallRRKMenu(Submenu); /* f() == (*f)() */
    FreeLibrary(hrrkmdll);
    return result;
}
```

Friends of MS's Basic dialects (f.e. Word-Basic or Visual Basic) can use following procedure:

```
Declare Function CallRRKMenu Lib "RRKMNDLL.DLL"(Submenu As String) As Integer
Sub MAIN
Submenu$ = "TOOLS"
REM calls the Submenu "Tools"
result = CallRRKMenu(Submenu$)
Print result
End Sub
```

Comments:

Please don't use `implib` or similar to link your C or Pascal program to RRKMNDLL.DLL, otherwise you may get problems with future versions of the Menu. Prefer the described method.

The return value of this function is by success TRUE.

## Assoziare a Window-Class with a Menu

New Menu for Windows is not only a Windows-Shell, but has some qualities to enhance Windows. If you can write a DLL and/or you know some details of Windows-Messages, you can customise and enhance the Windows-functionality of Windows standard elements (like EditBox, ListBox or Window-Classes of your application) with a Popup Menu. If you want participate to New Menu for Windows with your application, you haven't change any line of your code.

How this function, I demonstrated with the Window Classes EditBox, ListBox, ScroolBar an tty (the DOS-Box in the 386-modus).

If in the dialogue General Properties the option **Context-Sensitive Menus** is activated, the Window Classes, listed in the section `[WinClasses]`, are customised with a local Menu. This Menu can be popuped, if the mouse is over an registered class-window with the right mousebutton.

In the file **ADDONS.WRI** the defined context-sensitive menus are described.

In the command line of a menuitem you can call via MNU\_CALLDLL any function of DLL. You can use the expression `@HWND` for the Handle of class window as a parameter to a function.

But if you only want to send a message (via `PostMessage(...)`) to this window you can use the internal command MNU\_MTW.

It is also possible to send keyboard events to this window with MNU\_KB.

If you have some ideas to enhance Windows functionality or if you want use Menus for your applications, please contact me.

## Send Messages to a Window

With the internal command **MNU\_MTW** (Message to window) in a command line of a Windows Class Menu you can send Message to the Class-Window. (Of course it would also function with: MNU\_CALLDLL USER POSTMESSAGE @HWND (and so on.)).

MNU\_MTW Parent message wParam lParam

- MNU\_MTW sends the message with `PostMessage` to the **Parent**<sup>^</sup> of the Window. This means if `Parent=0`, the message will be send to `@HWND` (the Class Window). If `Parent=1` the message will be send to `GetParent (@HWND)`, if `Parent=2` to `GetParent (GetParent (@HWND))` and so on.
- **message** is the message to the window (almost `WM_`- constants) as `UINT`.
- **wParam** is the first parameter of the message as `WORD`.
- **lParam** is the second parameter of the message as `LONG`. (Exceptionally you haven't to sign this `LONG` with a 'L' after the number).

You can also declare **message**, **wParam** and **lParam** as hexadecimal numbers (f.e.: `0x1234`).

## Stick normal Drop-Down-Menus with a Pin

This feature is only available, if in the dialogue General Properties the option **HookMenus** is activated. If you select a Menu of a normal program with a menu bar not with left, but with right mousebutton, instead of the normal menu a Menu know by **New Menu for Windows** appears.

F.e.: If you select the menu 'File' of Write with the right mousebutton, you can stick this menu with a needle on the desktop. The sticked Menu is hold in the foreground every time. But if you change the focus to an other application the sticked Menus of Write would be hidden.

### **Attention:**

The Menus of an application can be changed (f.e. the menu 'Window'), so the sticked Menus are not up to date and send wrong commands to the application. This maybe cause unexpectable reactions and effects. The menuitems, which are not accessible (normally showed greyed) have no special representation in the sticked Menus. Anyway this feature is not very outperformed!

