

## **PopupMenu**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> PopupMenu	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		July 31, 2024

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>PopupMenu</b>	<b>1</b>
1.1	PopupMenu	1
1.2	attachpopupmenu	1
1.3	disablepopupmenuitem	2
1.4	freepopupmenu	2
1.5	initpopupmenu	3
1.6	popupmenubar	3
1.7	popupmenuchecked	3
1.8	popupmenucheckitem	3
1.9	popupmenuchecksubitem	4
1.10	popupmenuevent	4
1.11	popupmenuinfo	5
1.12	popupmenuitem	5
1.13	popupmenubar	6
1.14	popupmenubarinfo	6
1.15	popupmenubaritem	6
1.16	popupmenubarsubmenuitem	7
1.17	popupmenutitle	7
1.18	waitpopupmenuevent	7

## Chapter 1

# PopupMenu

### 1.1 PopupMenu

PureBasic - PopupMenu V1.00

A 'popup menu' is a quick menu which appears when you click the right mouse button on a particular window. In PureBasic, you can attach one popup menu per window which allow you to do context sensitive menu. The nice `popupmenu.library` is used to achieve that, so if you decide to use these functions, doesn't forget to include this library in your final package, as it's not a 'standard' library.

Commands summary:

```
AttachPopupMenu
DisablePopupMenuItem
FreePopupMenu
InitPopupMenu
PopupMenuBar
PopupMenuChecked
PopupMenuCheckItem
PopupMenuCheckSubItem
PopupMenuEvent
PopupMenuInfo
PopupMenuItem
PopupMenuSubBar
PopupMenuSubInfo
PopupMenuSubItem
PopupMenuSubMenuItem
PopupMenuTitle
WaitPopupMenuEvent
```

PopupMenu demo

### 1.2 attachpopupmenu

SYNTAX

```
Result.w = AttachPopupMenu (#Menu.w, Window.l)
```

---

**FUNCTION**

This function must be called as the last one after each menu building sequence, of commands like `PopupMenuTitle()`, `PopupMenuItem()`, `PopupMenuSubItem()` and so on, else the menu will not be completed.

**#Menu**

The menu to attach.

**Window**

A valid window pointer. Get it from either `OpenWindow()` or from `WindowID()`.

**Result**

If this is `TRUE` there have been one or more errors caught when the menu was builded, it could be possible to display it but don't expect it to look like it was intended, but if it's `FALSE` no error has occurred.

### 1.3 disablepopupmenuitem

**SYNTAX**

`Result.w = DisablePopupMenuItem(#Menu.w, Item.w, State.w)`

**FUNCTION**

To enable or disable any item in a popupmenu that could be selected.

**#Menu**

The menu to use.

**Item**

The item to enable or disable.

**State**

Set this to `FALSE` for enable and `TRUE` for disable.

**Result**

This is `TRUE` if the item is successfully changed else it's `FALSE`.

### 1.4 freepopupmenu

**SYNTAX**

`FreePopupMenu(#Menu.w)`

**STATEMENT**

Use this statement to free a popupmenu.

**#Menu**

The menu to free.

---

## 1.5 initpopupmenu

### SYNTAX

```
Result.l = InitPopupMenu(#Menus.l)
```

### FUNCTION

Call this function as the first one, as it is the init routine, and test the result if you want to be sure.

### #Menus

How many menus that is wanted.

### Result

If it's TRUE then all is fine but if it's instead FALSE then no other functions should be called.

## 1.6 popupmenubar

### SYNTAX

```
PopupMenuBar()
```

### STATEMENT

To have a nice separator bar in the popupmenu then call this statement.

## 1.7 popupmenuchecked

### SYNTAX

```
Result.w = PopupMenuChecked(#Menu.w, Item.w)
```

### FUNCTION

To see if a checkitem have the checkmark set or not.

### #Menu

The menu to use.

### Item

The item to look out for.

### Result

If this is FALSE then the checkmark is not set but if it's instead TRUE then it is.

## 1.8 popupmenucheckitem

### SYNTAX

```
PopupMenuCheckItem(Item.w, Text$, ShortCut$)
```

### STATEMENT

To create a checkitem for the popupmenu then call this statement. The checkmark will be in false state, not set.

Item

This must be a special number for each of the different items that is created, so when `WaitPopupMenuEvent()` or `PopupMenuEvent()` catch a popupmenu event this number is what they will return.

\* Se above functions for special note.

Text\$

The text for this checkitem.

ShortCut\$

This little one character string specify what the keyboard shortcut will be for this checkitem. If no shortcut will be used then supply a empty string.

## 1.9 popupmenuchecksubitem

SYNTAX

```
PopupMenuCheckSubItem(SubItem.w, Text$, ShortCut$)
```

STATEMENT

To create a subcheckitem for the popupmenu then call this statement. The checkmark will be in false state, not set.

SubItem

This must be a special number for each of the different items that is created so when `WaitPopupMenuEvent()` or `PopupMenuEvent()` catch a popupmenu event this number is what they will return.

\* Se above functions for special note.

Text\$

The text for this subcheckitem.

ShortCut\$

This little one character string specify what the keyboard shortcut will be for this item. If no shortcut will be used then supply a empty string.

## 1.10 popupmenuevent

SYNTAX

```
Result.l = PopupMenuEvent(#Menu.w)
```

FUNCTION

This function check if any event has occurred.

---

It is the right one to use if the event loop should go on and not do the wait that WaitPoupMenuEvent() will do.

#Menu  
The popupmenu to open if RMB is pressed.

Result  
The event that occurred.

\* Special Note.

It's not only the popupmenu event that will be returned, also any window event will be returned if any have taken place instead of a popupmenu event. It's a good idea to avoid itemnumber that is the same as window events.

## 1.11 popupmenuinfo

SYNTAX

PopupMenuInfo(Text\$)

STATEMENT

To create an infoitem use this statement. It will not be possible to select this item.

If the infoitem should be complete then a separator bar must be the next item after this else the text will be black and not the shiny one.

How a infoitem will be drawn are set with the prefs program that came along with the popupmenu.library.

Text\$

The text for this infoitem.

## 1.12 popupmenuitem

SYNTAX

PopupMenuItem(Item.w, Text\$, ShortCut\$)

STATEMENT

Use this statement to create a item for the popupmenu.

Item

This must be a special number for each of the different items that is created so when WaitPopupMenuEvent() or PopupMenuEvent() catch a popupmenu event this number is what they will return.

\* Se above functions for special note.

Text\$

---

The text for this item.

ShortCut\$

This little one character string specify what the keyboard shortcut will be for this item. If no shortcut will be used then supply a empty string.

## 1.13 popupmenusubbar

SYNTAX

PopupMenuSubBar()

STATEMENT

To have a nice separator bar in a submenu then use this statement.

## 1.14 popupmenusubinfo

SYNTAX

PopupMenuSubInfoItem(Text\$)

STATEMENT

To create an subinfoitem use this statement. It will not be possible to select this item.

If the subinfoitem should be complete then a separator bar must be the next item after this else the text will be black and not the shiny one.

How a infoitem will be drawn are set with the prefs program that came along with the popupmenu.library.

Text\$

The text for this infoitem.

## 1.15 popupmenusubitem

SYNTAX

PopupMenuSubItem(SubItem.w, Text\$, ShortCut\$)

STATEMENT

Use this statement to create a subitem for the popupmenu.

SubItem

This must be a special number for each of the different items that is created so when `WaitPopupMenuEvent()` or `PopupMenuEvent()` catch a popupmenu event this number is what they will return.

\* Se above functions for special note.

---

Text\$

The text for this subitem.

ShortCut\$

This little one character string specify what the keyboard shortcut will be for this item. If no shortcut will be used then supply a empty string.

## 1.16 popupmenusubmenuitem

SYNTAX

PopupMenuSubMenuItem(Text\$)

STATEMENT

Use this statement to create a entry to a submenu in the popupmenu.

All the different subitem statements will attach a item to this previously created submenu.

Text\$

The text for this submenuitem.

## 1.17 popupmenutitle

SYNTAX

PopupMenuTitle(Text\$)

STATEMENT

To create the menutitle then use this statement. It's the first menu statement that must be called when a popupmenu is about to be builded.

No separator bar is needed to be the next item after this one to have the nice and shiny text, as it have to be with PopupMenuInfo() and PopupMenuSubInfo().

Text\$

The text for the title.

## 1.18 waitpopupmenuevent

SYNTAX

Result.l = WaitPopupMenuEvent(#Menu.w)

FUNCTION

This function check if any event has occurred.

It is the one to use if there is no need for the

---

event loop to go on.

#Menu

The popupmenu to open if RMB is pressed.

Result

The event that occurred.

\* Special Note.

It's not only the popupmenu event that will be returned, also any window event will be returned if any have taken place instead of a popupmenu event. It's a good idea to avoid itemnumber that is the same as window events.