

PopupMenu

COLLABORATORS

	<i>TITLE :</i> PopupMenu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 23, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PopupMenu	1
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	4
1.9	popupmenuchecksubitem	4
1.10	popupmenuevent	5
1.11	popupmenuinfo	5
1.12	popupmenuitem	5
1.13	popupmenusubbar	6
1.14	popupmenusubinfo	6
1.15	popupmenusubitem	6
1.16	popupmenusubmenuitem	7
1.17	popupmenutitle	7
1.18	waitpopupmenuevent	8

Chapter 1

PopupMenu

1.1 PopupMenu

PureBasic - PopupMenu V1.00

Ein 'PopUp-Menü' ist ein Aufklapp-Menü, welches erscheint, sobald Sie mit der rechten Maustaste in einem entsprechenden Fenster klicken. In PureBasic können Sie ein PopUp-Menü an jedes Fenster anhängen, was Ihnen somit kontext-sensitive Menüs ermöglicht. Für diese Funktionen wird die `popupmenu.library` benutzt; wenn Sie also deren Funktionen nutzen möchten, vergessen Sie nicht, diese in Ihr endgültiges Programmpaket aufzunehmen, da es keine 'Standard' Library ist.

Befehlsübersicht:

- 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

Diese Funktion muß als letzte nach jeder Sequenz mit Menü-Befehlen wie `PopupMenuTitle()`, `PopupMenuItem()`, `PopupMenuSubItem()` usw. folgen, andernfalls wird das Menü nicht komplettiert.

#Menu

Das anzufügende Menü.

Window

Ein gültiger Window-Pointer. Ermitteln Sie ihn entweder mit `OpenWindow()` oder mit `WindowID()`.

Result

Ist dieses TRUE (wahr), dann wurden beim Menü-Aufbau ein oder mehrere Fehler bemerkt. Es ist möglich, dass das Menü angezeigt wird, aber erwarten Sie nicht, dass es wie beabsichtigt aussieht. Ist das Ergebnis FALSE (falsch) trat kein Fehler auf.

1.3 disablepopupmenuitem

SYNTAX

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

FUNCTION

Aktivieren oder deaktivieren eines Eintrages im PopUp-Menü, welcher ausgewählt werden kann.

#Menu

Das zu benutzende Menü.

Item

Der zu aktivierende oder zu deaktivierende Eintrag.

State

Setzen Sie den Status auf FALSE zum deaktivieren und auf TRUE zum aktivieren des Eintrags.

Result

Das Ergebnis ist TRUE, wenn der Eintrag erfolgreich geändert werden konnte, andernfalls ist es FALSE.

1.4 freepopupmenu

SYNTAX

```
FreePopupMenu(#Menu.w)
```

STATEMENT

Benutzen Sie diesen Befehl, um ein PopUp-Menü freizugeben.

#Menu

Das freizugebende Menü.

1.5 initpopupmenu

SYNTAX

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

FUNCTION

Rufen Sie diese Funktion als erste auf, da sie die Initialisierungs-Routine darstellt. Testen Sie auch das Ergebnis, wenn Sie diesbezüglich sichergehen wollen.

#Menus

Anzahl der gewünschten Menüs.

Result

Ist das Ergebnis TRUE ist alles in Ordnung, ist es aber FALSE sollten keine weiteren Funktionen aufgerufen werden.

1.6 popupmenubar

SYNTAX

```
PopupMenuBar()
```

STATEMENT

Um einen schönen Trennstrich im PopUp-Menü zu erhalten, rufen Sie diesen Befehl auf.

1.7 popupmenuchecked

SYNTAX

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

FUNCTION

Ermittelt, ob ein Checkitem mit einer Kontrollmarke gesetzt wurde oder nicht.

#Menu

Das zu benutzende Menü.

Item

Der zu überprüfende Eintrag (Item).

Result

Ist das Ergebnis FALSE, ist die Kontrollmarke nicht gesetzt. Ist es dagegen TRUE, ist sie gesetzt.

1.8 popupmenucheckitem

SYNTAX

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

STATEMENT

Um einen Eintrag mit Kontrollmarke (CheckItem) für ein PopUp-Menü zu erstellen, muß dieser Befehl aufgerufen werden. Die Kontrollmarke besitzt den Status FALSE, d.h. nicht gesetzt.

Item

Dies muß eine spezielle Nummer für jeden der verschiedenen Einträge sein, die erstellt wurden. Wenn dann WaitPopupMenuEvent() oder PopupMenuEvent() ein PopUp-Menü Ereignis überprüfen, wird die hier vergebene Nummer zurückgegeben.

* Siehe die vorhergehenden Funktionen für spezielle Anmerkungen.

Text\$

Der Text für dieses CheckItem.

ShortCut\$

Dieser kleine Ein-Zeichen-String gibt den Tastatur-Shortcut an, welcher für dieses CheckItem benutzt wird. Wenn kein Shortcut benutzt werden soll, dann übergeben Sie einen leeren String.

1.9 popupmenuchecksubitem

SYNTAX

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

STATEMENT

Um einen Untereintrag mit Kontrollmarke (CheckItem) für ein PopUp-Menü zu erstellen, muß dieser Befehl aufgerufen werden. Die Kontrollmarke besitzt den Status FALSE, d.h. nicht gesetzt.

SubItem

Dies muß eine spezielle Nummer für jeden der verschiedenen Einträge sein, die erstellt wurden. Wenn dann WaitPopupMenuEvent() oder PopupMenuEvent() ein PopUp-Menü Ereignis überprüfen, wird die hier vergebene Nummer zurückgegeben.

* Siehe die vorhergehenden Funktionen für spezielle Anmerkungen.

Text\$

Der Text für dieses SubCheckItem.

ShortCut\$

Dieser kleine Ein-Zeichen-String gibt den Tastatur-Shortcut an, welcher für dieses CheckItem benutzt wird. Wenn kein Shortcut benutzt werden soll, dann übergeben Sie einen leeren String.

1.10 popupmenuevent

SYNTAX

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

FUNCTION

Diese Funktion überprüft, ob irgendein Ereignis passierte.

Es ist die richtige Funktion, wenn die Ereignis-Schleife weiterlaufen soll und nicht warten soll, wie dies bei `WaitPopupMenuEvent()` geschieht.

#Menu

Das zu öffnende PopUpMenu, wenn die rechte Maustaste gedrückt wird.

Result

Das Ereignis (Event), welches stattfand.

* Spezielle Anmerkung:

Es wird nicht nur das PopUpMenu-Ereignis zurückgegeben, sondern auch jedes andere Window-Ereignis, falls irgendwelche anstelle eines PopUpMenu Ereignis stattfanden. Es ist eine gute Idee, Item-Nummern zu vermeiden, die dieselben wie die Window-Ereignisse sind.

1.11 popupmenuinfo

SYNTAX

```
PopupMenuInfo (Text$)
```

STATEMENT

Um ein InfoItem zu erstellen, benutzen Sie diesen Befehl. Es ist nicht möglich, diesen Eintrag auszuwählen.

Soll das InfoItem komplett sein, dann muß ein Separator (Trennbalken) der nächste Eintrag sein. Andernfalls wird der Text schwarz und nicht leuchtend dargestellt.

Wie ein InfoItem gezeichnet wird, wird mit dem Voreinstellungs-Programm eingestellt, welches mit der `popupmenu.library` mitgeliefert wird.

Text\$

Der Text für dieses InfoItem.

1.12 popupmenuitem

SYNTAX

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

STATEMENT

Benutzen Sie diesen Befehl, um einen Eintrag (Item) für das PopUpMenu zu erstellen.

Item

Dies muß eine spezielle Nummer für jeden der verschiedenen Einträge sein, die erstellt wurden. Wenn dann `WaitPopupMenuEvent()` oder `PopupMenuEvent()` ein PopUp-Menü Ereignis überprüfen, wird die hier vergebene Nummer zurückgegeben.

* Siehe die vorhergehenden Funktionen für spezielle Anmerkungen.

Text\$

Der Text für diesen Eintrag.

ShortCut\$

Dieser kleine Ein-Zeichen-String gibt den Tastatur-Shortcut an, welcher für dieses CheckItem benutzt wird. Wenn kein Shortcut benutzt werden soll, dann übergeben Sie einen leeren String.

1.13 popupmenusubbar

SYNTAX

`PopupMenuSubBar()`

STATEMENT

Um einen hübschen Trennbalken im Untermenü zu erhalten, benutzen Sie diesen Befehl.

1.14 popupmenusubinfo

SYNTAX

`PopupMenuSubInfoItem(Text$)`

STATEMENT

Um ein SubInfoItem (Info-Eintrag im Untermenü) zu erstellen, benutzen Sie diesen Befehl. Es ist nicht möglich, diesen Eintrag auszuwählen.

Soll das SubInfoItem komplett sein, dann muß ein Separator (Trennbalken) der nächste Eintrag sein. Andernfalls wird der Text schwarz und nicht leuchtend dargestellt.

Wie ein InfoItem gezeichnet wird, wird mit dem Voreinstellungs-Programm eingestellt, welches mit der `popupmenu.library` mitgeliefert wird.

Text\$

Der Text für dieses SubInfoItem.

1.15 popupmenusubitem

SYNTAX

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

STATEMENT

Benutzen Sie diesen Befehl, um einen Untereintrag für das PopUpMenu zu erstellen.

SubItem

Dies muß eine spezielle Nummer für jeden der verschiedenen Einträge sein, die erstellt wurden. Wenn dann `WaitPopupMenuEvent()` oder `PopupMenuEvent()` ein PopUp-Menü Ereignis überprüfen, wird die hier vergebene Nummer zurückgegeben.

* Siehe die vorhergehenden Funktionen für spezielle Anmerkungen.

Text\$

Der Text für dieses SubItem (Untereintrag).

ShortCut\$

Dieser kleine Ein-Zeichen-String gibt den Tastatur-Shortcut an, welcher für dieses CheckItem benutzt wird. Wenn kein Shortcut benutzt werden soll, dann übergeben Sie einen leeren String.

1.16 popupmenusubmenuitem

SYNTAX

`PopupMenuSubMenuItem(Text$)`

STATEMENT

Benutzen Sie diesen Befehl, um einen Eintrag im Untermenü (SubMenu) des PopUpMenu zu erstellen.

Alle der verschiedenen SubItem Befehle fügen einen Eintrag an dieses zuvor erstellte Submenu an.

Text\$

Der Text für diesen Submenu-Eintrag.

1.17 popupmenutitle

SYNTAX

`PopupMenuTitle(Text$)`

STATEMENT

Um einen Menü-Titel zu erstellen, benutzen Sie diesen Befehl. Es ist der erste Menü-Befehl, der aufgerufen werden muß, wenn ein PopUpMenu erstellt werden soll.

Um den hübschen leuchtenden Text zu erhalten, wird hier kein Trennbalken benötigt, wie dies noch bei `PopupMenuInfo()` und `PopupMenuSubInfo()` notwendig war.

Text\$
Der Text für den Titel.

1.18 waitpopupmenuevent

SYNTAX

Result.l = WaitPopupMenuEvent (#Menu.w)

FUNCTION

Diese Funktion überprüft, ob irgendein Ereignis stattfand.

Diese Funktion wird benutzt, wenn kein Bedarf für eine fortlaufende Ereignis-Schleife besteht.

#Menu

Das zu öffnende PopUpMenu, wenn die rechte Maustaste gedrückt wird.

Result

Das Ereignis, das stattfand.

* Spezielle Anmerkung:

Es wird nicht nur das PopUpMenu-Ereignis zurückgegeben, sondern auch jedes andere Window-Ereignis, falls irgendwelche anstelle eines PopUpMenu Ereignis stattfanden. Es ist eine gute Idee, Item-Nummern zu vermeiden, die dieselben wie die Window-Ereignisse sind.
