

帮助主题

主菜单

帕维尔, 2012 年 1 月 31 日 (创建于 2011 年 11 月 14 日)

PUM - 弹出式菜单
WinClip

无标签

PUM - 弹出菜单

帕维尔, 2012 年 1 月 31 日 (创建于 2011 年 11 月 14 日)

PUM 是一个专为创建和使用可自定义自绘弹出菜单而开发的 [AutoHotkey_L](#) 模块

无标签

[下载链接 \(.zip 13.3 KB\)](#)

包含三个用于创建和操作菜单的类:

PUM - 主管理类, 负责创建菜单
PUM_Menu - 表示弹出菜单的类
PUM_Item - 表示弹出菜单项的类

使用示例:

代码:

```
#NoEnv
#include PUM_API.ahk ;include this first
#include PUM.ahk

; parameters of the PUM object, the manager of the menus
pumParams := { "SelMethod" : "fill"           ;item selection method, may be frame,fill
               ;~ ,"selTColor" : -1           ;selection text color
               ;~ ,"selBGColor" : -1          ;selection background color, -1 means invert current color
               ,"oninit" : "PUM_out"          ;function which will be called when any menu going to be opened
               ,"onuninit" : "PUM_out"        ;function which will be called when any menu going to be closing
               ,"onselect" : "PUM_out"        ;;function which will be called when any item selected with mouse (hovered)
               ,"onrbutton" : "PUM_out"       ;function which will be called when any item right clicked
               ,"onmbbutton" : "PUM_out"      ;function which will be called when any item clicked with middle mouse button
               ,"onlbutton" : "PUM_out"       ;function which will be called when any item clicked with left mouse button
               ,"onshow" : "PUM_out"          ;function which will be called before any menu shown using Show method
               ,"onclose" : "Pum_out"         ;function called just before quitting from Show method
               ,mnemonicCMD : "select"}

;PUM_Menu parameters
menuParams1 := { "bgcolor" : 0x36311f        ;background color of the menu
                 , "iconssize" : 32          ;size of icons in the menu
                 , "tcolor" : 0xafc9d3 }     ;text color of the menu items
menuParams2 := { "bgcolor" : 0x1C3150
                 , "iconssize" : 16
                 , "tcolor" : 0xFFFFFF }

;create an instance of PUM object, it is best to have only one of such in the program
pm := new PUM( pumParams )

;creating popup menu, represented by PUM_Menu object with given parameters
menu := pm.CreateMenu( menuParams1 )

;create a three othe menus
newmenu1 := pm.CreateMenu( menuParams2 )
newmenu2 := pm.CreateMenu( menuParams2 )
newmenu3 := pm.CreateMenu( menuParams2 )

;adding submenu items to the first menu, item SubItem1 will open "newmenu1" menu, and SubItem2 will open "newmenu2"
menu.add( { "name" : "SubItem1", "submenu" : newmenu1, "icon" : "shell32.dll:8" } )
menu.add( { "name" : "SubItem2", "submenu" : newmenu2, "icon" : "shell32.dll:8" } )
menu.Add() ;adding separator
;adding a submenu item "SubItem3" to the "newmenu2" menu, which opens "newmenu3" menu
newmenu2.add( { "name" : "SubItem3", "icon" : "shell32.dll:8", "submenu" : newmenu3 } )

;adding five items to the first menu
loop, 5
    menu.Add( { "name" : "i&tem" A_Index
               , "bold" : 1
               , "icon" : "shell32.dll:" A_index+20 } )
;adding five items to the newmenu1
```

```
loop,5
    newmenu1.add( { "name" : "item" A_index, "icon" : "shell32.dll:3" A_index } )
;adding five items to the newmenu2
loop,5
    newmenu2.add( { "name" : "item" A_index, "icon" : "shell32.dll:4" A_index } )
;adding five items to the newmenu3
loop,5
    newmenu3.add( { "name" : "item" A_index, "icon" : "shell32.dll:2" A_index, disabled : 1 } )

;showing the first menu at the center of screen (~)
if ( item := menu.Show( A_ScreenWidth/3, A_ScreenHeight/3 ) )
    msgbox % "Chooosen item: " item.name

;Destroying all menus/items created
;Use Destroy() method for the PUM_Menu object if you want to destroy specific menu
pm.Destroy()
ExitApp
return

PUM_out( msg, obj )
{
    if ( msg = "onselect" )
    {
        rect := obj.GetRECT()
        CoordMode, ToolTip, Screen
        tooltip,% "Selected: " obj.name,% rect.right,% rect.top
    }
    if ( msg ~= "oninit|onuninit|onshow|onclose" )
        tooltip % "menu " msg ": " obj.handle
    if ( msg = "onrbutton" )
        tooltip % "Right clicked: " obj.name
    if ( msg = "onmbutton" )
        tooltip % "Middle clicked: " obj.name
    if ( msg = "onrun" )
        tooltip % "Item runned: " obj.name
}
```

PUM_项目

帕维尔，2012 年 1 月 31 日 （创建于 2011 年 11 月 15 日）

PUM_Item 是通过 PUM_Menu.Add() 方法创建的弹出菜单项类
一个项目可以关联菜单作为子菜单

无标签

示例：

```
ItemParams := { "name" : "SomeItemName"
                , "bold" : 1
                , "icon" : "shell32.dll:20" }

oPUM := new PUM()
menu := oPUM.CreateMenu( MenuParams )
item := menu.Add( ItemParams )
menu.Show( 100, 100, "hcenter animlr" )
item.Destroy()
```

PUM_项参数。请仅通过 SetParams() 方法修改这些参数

参数名称	默认值	可能的值	描述
issep	0	0,1	1 表示是分隔符, 0 表示不是
名称	<item_title>	字符串	物品名称
粗体	0	0,1	若为1, 该物品将在菜单中以粗体显示
图标	0	字符串	图标路径或句柄, 可采用路径:图标索引格式, 支持任意图片文件
换行	0	0,1,2	若为1 - 将在此项目后中断菜单, 下一项将开启新列 如果 2 - 菜单分隔符将以垂直线形式显示
子菜单	0	PUM_Menu 对象 或弹出菜单句柄	如果存在, 悬停时项目将打开子菜单
tcolor	""	RGB	项目文本颜色, 若为空则使用父菜单的颜色
背景颜色	""	RGB	项目背景色, 若为空则使用父菜单颜色
无前缀	0	0,1	如果为1, 则不解释"&"助记符
用户 ID	""	字符串	该物品的自定义 ID, 可以是任意字符串或整数
已禁用	0	0,1	如果为1, 物品将显示为禁用状态且不可点击
图标使用句柄	0	0,1	如果此标志为 True, 且通过 SetParams 在"icon"字段中传递了图标句柄, 则删除物品时不会销毁此句柄

只读字段，请勿修改：

名称	描述
存活	若物品被摧毁则为非1
标识符	物品的内部标识符，始终保持唯一性
菜单	PUM_Menu 对象，此项目所属
关联菜单	PUM_Menu 对象，作为该项目的子菜单关联
hFont	该项目字体的句柄
hlcon	与商品图标关联的句柄
热键字符代码	菜单打开时用户按下任意键需检测的字符 ASCII 码

PUM_Item 方法：

名称	输入	输出	描述
SetParams()	objItemParams	-	更新物品的必需参数，仅更新 objItemParams 中存在的参数
RemoveSubMenu()	-	-	分离与项目菜单的关联，不会销毁项目或菜单
销毁子菜单()	-	-	销毁与物品菜单相关的项目
获取位置()	-	从0开始位置	返回此项在菜单中从0开始的位置
获取矩形区域	-	对象	返回表示当前菜单项边界矩形的 RECT 对象，该矩形以屏幕坐标表示 对象包含以下字段：left（左）、right（右）、top（上）、bottom（下）
更新()	-	-	更新物品参数。当您直接修改了参数而未使用 SetParams() 方法时特别有用
销毁()	-	-	销毁物品及关联菜单（如果存在）
分离()	-	-	销毁物品并移除菜单而不销毁物品本身
获取文本颜色()	-	RGB	返回此物品用作文本颜色的 RGB 颜色值
获取背景颜色()	-	RGB	返回此项目使用的 RGB 颜色值作为背景色
获取图标句柄()	-	hlcon	返回此项目用于绘制图标的图标句柄

PUM_Menu

[帕维尔](#)，2012 年 1 月 31 日（创建于 2011 年 11 月 15 日）

[PUM_Menu](#) 是通过 [PUM.CreateMenu\(\)](#) 方法创建的弹出菜单类
菜单可与商品关联，此时它将有表示所属 [PUM_Item](#) 对象的“所有者”字段
销毁 [PUM_Menu](#) 时，其包含的所有商品也将被销毁

无标签

示例：

```
MenuParams := { "bgcolor" : 0x36311f
                , "iconssize" : 16
                , "tcolor" : 0xafc9d3 }

oPUM := new PUM()
menu := oPUM.CreateMenu( MenuParams )
menu.Add( "item" )
menu.Show( 100, 100, "hcenter animlr" )
menu.Destroy()
```

PUM_菜单参数。请仅通过 [SetParams\(\)](#) 方法修改这些参数

参数名称	默认值	可能的值	描述
tcolor	windows 默认值	RGB	菜单项文本颜色
背景颜色	Windows 默认设置	RGB	菜单的 RGB 背景色
无颜色	0	0,1	如果为1，将使用默认颜色作为菜单背景和商品文字颜色
无图标	0	0,1	如果设为1，菜单中将不显示图标
无文本	0	0,1	如果为 1，将不显示该项目的文本，不应与“noicons”同时使用
图标尺寸	32	整数 最好为16或32	项目图标尺寸
文本偏移量	5	整数	图标与项目文本之间的像素间距
最大高度	0	int	菜单的最大高度，若菜单内容超出将添加滚动条
左右边距	3	整数	项目边界的左右边距
垂直边距	3	int	商品边界的上下边距
文本边距	5	int	文本后添加的像素数量，使菜单看起来更美观

只读字段，请勿修改：

名称	描述
存活	若菜单被销毁则不为1
句柄	弹出菜单的句柄
objPUM	对父级 PUM 对象的引用
所有者	对 PUM_Item 对象的引用（如果存在与此菜单关联的对象）

PUM_菜单方法：

名称	进入	退出	描述
SetParams()	objMenuParams	-	更新菜单的必要参数， 仅更新 objMenuParams 中存在的参数
添加()	objItemParams 上一个项目（可选） 按位置查找（可选）	PUM_项目 成功时 0 表示失败	向菜单添加新项目 objItemParams - 新项目的参数 prevItem - 新项目将添加于其后方的项目位置或 ID fByPos - 若为 1，prevItem 表示从 0 开始的位置索引；若为 0，prevItem 表示新项目将添加于其后方的 PUM_Item.id 要将项目添加到菜单末尾，请传递 -1 而非 prevItem（默认情况下） 如果 objItemParams 为空，将添加分隔符 如果 objItemParams 为任意字符串，将添加具有该名称的项目
GetItems()	-	PUM_项 数组	返回当前菜单所包含的条目数组
结束菜单()	-	-	立即关闭屏幕上所有弹出式菜单
IsMenu()	-	1 或 0	如果当前菜单仍然存在（未被销毁）则返回 1
GetItemByPos()	从0开始的位置	PUM_Item	返回表示菜单中特定位置 PUM_Item 对象的项目
Count()	-	项目数量 在菜单中	返回菜单中当前的项目数量
GetTColor()	-	RGB	返回此菜单当前使用的文本颜色
获取背景颜色()	-	RGB	返回此菜单当前使用的背景颜色

销毁()	-	-	销毁菜单及其所有菜单项，以及与此菜单项关联的所有菜单
分离()	-	-	将此菜单从其所属项目中分离。不会销毁任何内容
显示()	x y 标志	PUM_Item （若存在任何选中项） 0（其他情况）	在屏幕坐标 X,Y 处显示此菜单 若存在选中项，此函数将返回其 PUM_Item 对象 Flags 是以空格分隔的选项列表，选项如下所示

在 **PUM_Menu.Show()**中使用的标志：

名称	描述
上下文	将在当前打开的菜单上方显示上下文弹出菜单， 若在无其他菜单显示时使用此标志，菜单将不会显示
水平居中	将快捷菜单相对于 x 参数指定的坐标水平居中。
左侧对齐	（默认）将快捷菜单定位，使其左侧与 x 参数指定的坐标对齐。
右侧对齐	将快捷菜单定位，使其右侧与 x 参数指定的坐标对齐。
底部对齐	将快捷菜单定位，使其底边与 y 参数指定的坐标对齐。
vtop	（默认）将快捷菜单定位，使其顶部与 y 参数指定的坐标对齐。
vcenter	将快捷菜单相对于 y 参数指定的坐标垂直居中。
animlr	从左到右为菜单添加动画效果。
animrl	从右到左为菜单添加动画效果。
animtb	从上到下为菜单添加动画效果。
animbt	从下到上为菜单添加动画效果。
无动画	无动画显示菜单。

PUM

帕维尔，2012 年 1 月 31 日（创建于 2011 年 11 月 15 日）

PUM 是管理器类，通过它可以创建 **PUM_Menu** 菜单

程序中最好只保留一个 **PUM** 实例

销毁 **PUM** 对象时，通过该对象创建的所有菜单/项目也将被销毁。

无标签

示例：

```
PUMParams := { "SelMethod" : "fill"  
               , "selTColor" : -1  
               , "selBGColor" : -1}  
oPUMMenu := new PUM( PUMParams )  
menu := oPUMMenu.CreateMenu()  
menu.Add( "item" )  
menu.Show( 100, 100 )  
oPUMMenu.Destroy()
```

PUM 参数。仅通过 **SetParams()** 方法修改这些参数

参数名称	默认值	可能的值	描述
选择方法	填充	填充 框架	当用户将鼠标光标悬停在菜单项上时出现的一种选择形式
selBGColor	Windows 默认设置	任意 RGB	所选项的背景色，-1表示将反转当前颜色
selTColor	Windows 默认	任意 RGB 值	所选项的文本颜色，-1表示将反转当前颜色
边框宽度	1	任意整数	当 selMethod = "frame"时的选择框宽度
助记符命令	运行	选择 运行	指示当用户按下与菜单项关联的任何热键时会发生什么 "运行"表示该商品将如同用户点击一样被执行 "select" 表示该选项将被选中/高亮显示

参数名称	默认值	可能的值	描述
初始化时	""	任意有效的函数名称	函数名称，该函数将在任何菜单打开时运行 目标函数的第一个参数将等于"oninit" 第二个参数将是正在打开的菜单的 PUM_Menu 实例
onuninit	""	任意有效的函数名称	函数名称，该函数将在任何菜单关闭时运行 目标函数的第一个参数将等于"onuninit" 第二个参数将是正在关闭的菜单的 PUM_Menu 实例
onselect	""	任何有效的函数名称	函数名称，当菜单中任何项被选中或悬停时将运行该函数 目标函数的第一个参数将等于"onselect" 第二个参数将是选中项的 PUM_Item 实例
onrbutton	""	任意有效的函数名	函数名称，当用户在物品上按下鼠标右键时将运行该函数 目标函数的第一个参数将等于"onrbutton" 第二个参数将是目标项的 PUM_Item 实例
onmbutton	""	任何有效的函数名称	函数名称，当用户在项上按下鼠标中键时将运行该函数 目标函数的第一个参数将等于"onmbutton" 第二个参数将是目标项的 PUM_Item 实例
onrun	""	任意有效的函数名	函数名称，当用户运行任何商品时该函数将被执行 目标函数的第一个参数将等于"onrun" 第二个参数将是目标项的 PUM_Item 实例
显示时	""	任何有效的函数名称	函数的名称，该函数将在每次调用 PUM_Menu.Show() 方法后、菜单显示前执行 目标函数的第一个参数将是"onshow" 第二个参数将是即将显示的菜单的 PUM_Menu 实例 如果目标函数存在且返回值为0，菜单将不会显示
onclose	""	任意有效的函数名称	函数名称，该函数将在每次 PUM_Menu.Show() 方法调用后菜单关闭时运行 目标函数的第一个参数将是 "onclose" 第二个参数将是所显示菜单的 PUM_Menu 实例

PUM 方法

方法名称	输入参数	返回值	描述
设置参数()	objPUMParams	-	更新 PUM 对象的必需参数，仅更新 objParams 中存在的参数
获取菜单()	菜单句柄	PUM 菜单 操作成功 否则为 0	通过句柄检索 PUM_Menu 对象
GetItemByUID()	uid	PUM_Item 成功时返回 失败时返回 0	通过其 uid（用户自定义 ID）检索 PUM_Item 对象
GetItemByID()	id	成功时返回 PUM_Item 对象 失败时返回 0	通过商品 ID 获取 PUM_Item 对象（商品添加到菜单后可通过 PUM_Item.id 获取该 ID）
创建菜单()	菜单参数对象	弹出式菜单 成功时返回 失败时返回0	使用给定参数创建新的 PUM_Menu 对象
销毁()	-	-	销毁 PUM 对象及通过它创建的所有菜单/项