

## Popobject.mui

1. [Super class](#)
2. [Inherited by](#)
3. [Background](#)
4. [Attributes](#)
5. [MUIA Popobject Follow](#)
6. [MUIA Popobject Light](#)
7. [MUIA Popobject Object](#)
8. [MUIA Popobject ObjStrHook](#)
9. [MUIA Popobject StrObjHook](#)
10. [MUIA Popobject Volatile](#)
11. [MUIA Popobject WindowHook](#)

# Popobject.mui

## Super class

[Popstring.mui](#)

## Inherited by

[Poplist.mui](#)

[Popscreen.mui](#)

## Background

Popobject class takes a MUI object as parameter uses this one as popup. You can e.g. simply create a listview object with some entries and the popobject class will create a window around it and display it when the user hits the popup button.

Using this class instead of creating the popup windows yourself prevents you from having lots of problems. Think twice before deciding to make you own popups!

## Attributes

Attribute	Version	ISG	Type
<a href="#">MUIA Popobject Follow</a>	V7	ISG	BOOL
<a href="#">MUIA Popobject Light</a>	V7	ISG	BOOL
<a href="#">MUIA Popobject Object</a>	V7	I.G	Object *
<a href="#">MUIA Popobject ObjStrHook</a>	V7	ISG	struct Hook *
<a href="#">MUIA Popobject StrObjHook</a>	V7	ISG	struct Hook *
<a href="#">MUIA Popobject Volatile</a>	V7	ISG	BOOL
<a href="#">MUIA Popobject WindowHook</a>	V9	ISG	struct Hook *

## MUIA\_Popobject\_Follow

### NAME

MUIA\_Popobject\_Follow — V7 [ISG], `BOOL`, 0x80424cb5

### FUNCTION

Setting this attribute causes the popup window to follow its parent window when its moved.

Defaults to TRUE.

### SEE ALSO

MUIA\_Popobject\_Light, MUIA\_Popobject\_Volatile.

## MUIA\_Popobject\_Light

### NAME

MUIA\_Popobject\_Light — V7 [ISG], `BOOL`, 0x8042a5a3

### FUNCTION

This attribute causes the popup window to be border and titleless.

Defaults to TRUE

### SEE ALSO

MUIA\_Popobject\_Follow, MUIA\_Popobject\_Volatile

## MUIA\_Popobject\_Object

### NAME

MUIA\_Popobject\_Object — V7 [I.G], `Object *`, 0x804293e3

### FUNCTION

Specify the object to pop up. Usually this is a relatively simple thing like a single listview, but you can of course use group class here and make rather complex popups. As with all other MUI classes, the object here gets disposed when the popobject is disposed.

## EXAMPLE

```
pop PopobjectObject
  MUIA_Popstring_String KeyString
  MUIA_Popstring_Button PopButtonMUII_PopUp
  MUIA_Popobject_StrObjHook StrObjHook
  MUIA_Popobject_ObjStrHook ObjStrHook
  MUIA_Popobject_Object ListviewObject
    MUIA_Listview_List ListObject
      InputListFrame
      MUIA_List_SourceArray PopNames
    End
  End
End
```

## SEE ALSO

[MUIA\\_Popobject\\_StrObjHook](#), [MUIA\\_Popobject\\_ObjStrHook](#), [MUIA\\_Popobject\\_Light](#)

## MUIA\_Popobject\_ObjStrHook

### NAME

[MUIA\\_Popobject\\_ObjStrHook](#) — V7 [ISG], struct Hook \*, 0x8042db44

### FUNCTION

When a popup is closed, this hook is called. You can examine the state of your [MUIA\\_Popobject\\_Object](#) and set the contents of the string gadget respectively. The hook receives a pointer to itself in A0, a pointer to your [MUIA\\_Popobject\\_Object](#) in A2 and a pointer to the embedded string object in A1.

The hook will only be called when your popup is closed with a success value of TRUE. Otherwise, MUI closes the popup without taking further actions, just as if had never opened.

Since MUI doesn't know anything about your [MUIA\\_Popobject\\_Object](#), it's your task to tell when your popup is finished. You can terminate popups at any time by sending a [MUIM\\_Popstring\\_Close](#) method:

## EXAMPLE

```
DoMethodplist MUIM_Notify MUIA_Listview_DoubleClick TRUE
  pop
  MUIM_Popstring_Close TRUE

SAVEDS ASM VOID REGa2 Object listREGa1 Object str

  x

DoMethodplist MUIM_List_GetEntry MUIV_List_GetEntry_Active x
setstr MUIA_String_Contents x
```

# MUIA\_Popobject\_StrObjHook

## NAME

MUIA\_Popobject\_StrObjHook — V7 [ISG], struct Hook \*, 0x8042fbe1

## FUNCTION

Before the popup opens, this hook is called. You can use it to prepare your MUIA\_Popobject\_Object according to the contents of the string gadget. The hook receives a pointer to itself in A0, a pointer to your MUIA\_Popobject\_Object in A2 and a pointer to the embedded string object in A1.

Return TRUE if you want the popup to appear, FALSE otherwise.

## EXAMPLE

```
SAVEDS ASM LONG REGa2 Object list REGa1 Object str

    xs
    i

    getstr MUIA_String_Contents s

    ii

    DoMethodlist MUIM_List_GetEntry i x
    x

    setlist MUIA_List_Active MUIV_List_Active_Off

    stricmpx s

    setlist MUIA_List_Active i

returnTRUE
```

## SEE ALSO

MUIA\_Popobject\_ObjStrHook, MUIA\_Popobject\_Object, MUIA\_Popobject\_WindowHook

# MUIA\_Popobject\_Volatile

## NAME

MUIA\_Popobject\_Volatile — V7 [ISG], `BOOL`, 0x804252ec

## FUNCTION

Setting this attribute causes the popup window to disappear when the corresponding popobject disappears, e.g. because its in a page group and the user toggled the page. When the popobject appears again, the popup window appears also.

Defaults to TRUE.

## SEE ALSO

MUIA\_Popobject\_Light, MUIA\_Popobject\_Follow

# MUIA\_Popobject\_WindowHook

## NAME

MUIA\_Popobject\_WindowHook — V9 [ISG], `struct Hook *`, 0x8042f194

## FUNCTION

If specified, this hook is called immediately after the popup's window object has been created but before this window is opened. You might e.g. want to add a cycle chain for the popup window here.

The hook is called with a pointer to the pop object (MUIA\_Popobject\_Object) in A2 and with a pointer to the window object that MUI generated to handle the popup in A1.

## EXAMPLE

```
SAVEDS ASM VOID REGa2 Object popREGa1 Object win
    setwin MUIA_Window_DefaultObject pop
```

## SEE ALSO

MUIA\_Popobject\_ObjStrHook, MUIA\_Popobject\_Object

---

Copyright © 1992-2006 by Stefan  
Stuntz  
Copyright © 2006-2018 by Thore  
Böckelmann, Jens Maus

MUI for AmigaOS -  
MUI-Autodocs

Updated: 06-Dec-2018