

## Boopsi.mui

1. [Super class](#)
2. [Background](#)
3. [Attributes](#)
4. [MUIA Boopsi Class](#)
5. [MUIA Boopsi ClassID](#)
6. [MUIA Boopsi MaxHeight](#)
7. [MUIA Boopsi MaxWidth](#)
8. [MUIA Boopsi MinHeight](#)
9. [MUIA Boopsi MinWidth](#)
10. [MUIA Boopsi Object](#)
11. [MUIA Boopsi Remember](#)
12. [MUIA Boopsi Smart](#)
13. [MUIA Boopsi TagDrawInfo](#)
14. [MUIA Boopsi TagScreen](#)
15. [MUIA Boopsi TagWindow](#)

# Boopsi.mui

## Super class

[Gadget.mui](#)

## Background

MUI's Boopsi class provides an interface to standard, system style BOOPSI gadgets. Since BOOPSI's gadgetclass misses some important features needed for an automatic layout system like MUI, there are several problems with such an interface. MUI tries to solve these problems with some additional attributes.

Coming with release 3.x of the Amiga operating system are some very nice BOOPSI gadgets such as "colorwheel.gadget" or "gradientslider.gadget". With MUI's Boopsi class, you can use these gadgets just as if they were MUI objects.

You can talk to a MUIized BOOPSI object as if it was the BOOPSI object itself. MUI will pass through all attributes and try to be completely transparent. Additionally, if a BOOPSI object generates notification events via IDCMP\_UPDATE, MUI turns them into MUI notification events. Thus, you can e.g. react on the change of WHEEL\_Saturation in a MUI colorwheel BOOPSI gadget as on any other MUI attribute.

An example program "BoopsiDoor.c" is provided to show how this magic works.

## NOTES

OS 3.0/3.1 colorwheel.gadget can accidentally render itself one pixel too big, overwriting other parts of the window. As a workaround, MUI will subtract one from the width/height before passing it on to a colorwheel BOOPSI object.

# Attributes

Attribute	Version	ISG	Type
<u>MUIA_Boopsi_Class</u>	V4	ISG	struct IClass *
<u>MUIA_Boopsi_ClassID</u>	V4	ISG	STRPTR
<u>MUIA_Boopsi_MaxHeight</u>	V4	ISG	ULONG
<u>MUIA_Boopsi_MaxWidth</u>	V4	ISG	ULONG
<u>MUIA_Boopsi_MinHeight</u>	V4	ISG	ULONG
<u>MUIA_Boopsi_MinWidth</u>	V4	ISG	ULONG
<u>MUIA_Boopsi_Object</u>	V4	..G	Object *
<u>MUIA_Boopsi_Remember</u>	V4	I..	ULONG
<u>MUIA_Boopsi_Smart</u>	V9	I..	BOOL
<u>MUIA_Boopsi_TagDrawInfo</u>	V4	ISG	ULONG
<u>MUIA_Boopsi_TagScreen</u>	V4	ISG	ULONG
<u>MUIA_Boopsi_TagWindow</u>	V4	ISG	ULONG

## MUIA\_Boopsi\_Class

### NAME

MUIA\_Boopsi\_Class — V4 [ISG], struct IClass \*, 0x80426999

### FUNCTION

Pointer to the (private) class you want to create a BOOPSI object from. Only useful if you previously generated your own BOOPSI class with MakeClass().

Of course you may not free the class until you're done with your object.

### SEE ALSO

MUIA\_Boopsi\_ClassID

## MUIA\_Boopsi\_ClassID

### NAME

MUIA\_Boopsi\_ClassID — V4 [ISG], STRPTR, 0x8042bfa3

### FUNCTION

MUIA\_Boopsi\_ClassID specifies the name for the public BOOPSI class you want to create an object of. It will only be used when MUIA\_Boopsi\_Class is NULL.

The public class must be in memory before you can create an instance of it, you will have to open the required class library by hand.

Note the string given to MUIA\_Boopsi\_ClassID must remain valid until you're done with the object.

## EXAMPLE

```

cwbases OpenLibrary

Wheel BoopsiObject
  NeXTFrame
  MUIA_Boopsi_ClassID
  MUIA_Boopsi_MinWidth
  MUIA_Boopsi_MinHeight
  MUIA_Boopsi_Remember WHEEL_Saturation
  MUIA_Boopsi_Remember WHEEL_Hue
  MUIA_Boopsi_TagScreen WHEEL_Screen
  WHEEL_Screen
  GA_Left
  GA_Top
  GA_Width
  GA_Height
  ICA_TARGET ICTARGET_IDCMP
End

MUI_DisposeObjectwheel
CloseLibrarycwbases
```

## SEE ALSO

MUIA\_Boopsi\_Class

## MUIA\_Boopsi\_MaxHeight

### NAME

MUIA\_Boopsi\_MaxHeight — V4 [ISG], ULONG, 0x8042757f

### FUNCTION

For MUI's automatic layout system, it's required that objects know their minimum and maximum sizes. Since BOOPSI gadgets don't support this feature, you will have to help MUI and adjust these values by hand.

Defaults:

- MUIA\_Boopsi\_MinWidth - 1 pixel
- MUIA\_Boopsi\_MinHeight - 1 pixel
- MUIA\_Boopsi\_MaxWidth - unlimited
- MUIA\_Boopsi\_MaxHeight - unlimited

## EXAMPLE

see [MUIA\\_Boopsi\\_ClassID](#)

## SEE ALSO

[MUIA\\_Boopsi\\_ClassID](#)

# MUIA\_Boopsi\_MaxWidth

## NAME

[MUIA\\_Boopsi\\_MaxWidth](#) — V4 [ISG], ULONG, 0x8042bcb1

## FUNCTION

For MUI's automatic layout system, it's required that objects know their minimum and maximum sizes. Since BOOPSI gadgets don't support this feature, you will have to help MUI and adjust these values by hand.

Defaults:

[MUIA\\_Boopsi\\_MinWidth](#) - 1 pixel  
[MUIA\\_Boopsi\\_MinHeight](#) - 1 pixel  
[MUIA\\_Boopsi\\_MaxWidth](#) - unlimited  
[MUIA\\_Boopsi\\_MaxHeight](#) - unlimited

## EXAMPLE

see [MUIA\\_Boopsi\\_ClassID](#)

## SEE ALSO

[MUIA\\_Boopsi\\_ClassID](#)

# MUIA\_Boopsi\_MinHeight

## NAME

[MUIA\\_Boopsi\\_MinHeight](#) — V4 [ISG], ULONG, 0x80422c93

## FUNCTION

For MUI's automatic layout system, it's required that objects know their minimum and maximum sizes. Since BOOPSI gadgets don't support this feature, you will have to help MUI and adjust these values by hand.

EXAMPLE

Defaults:

MUIA\_Boopsi\_MinWidth - 1 pixel  
MUIA\_Boopsi\_MinHeight - 1 pixel  
MUIA\_Boopsi\_MaxWidth - unlimited  
MUIA\_Boopsi\_MaxHeight - unlimited

## EXAMPLE

see MUIA\_Boopsi\_ClassID

## SEE ALSO

MUIA\_Boopsi\_ClassID

# MUIA\_Boopsi\_MinWidth

## NAME

MUIA\_Boopsi\_MinWidth — V4 [ISG], ULONG, 0x80428fb2

## FUNCTION

For MUI's automatic layout system, it's required that objects know their minimum and maximum sizes. Since BOOPSI gadgets don't support this feature, you will have to help MUI and adjust these values by hand.

Defaults:

MUIA\_Boopsi\_MinWidth - 1 pixel  
MUIA\_Boopsi\_MinHeight - 1 pixel  
MUIA\_Boopsi\_MaxWidth - unlimited  
MUIA\_Boopsi\_MaxHeight - unlimited

## EXAMPLE

see MUIA\_Boopsi\_ClassID

## SEE ALSO

MUIA\_Boopsi\_ClassID

# MUIA\_Boopsi\_Object

FUNCTION

## NAME

MUIA\_Boopsi\_Object — V4 [..G], Object \*, 0x80420178

## FUNCTION

No input, just an output since this attribute is only gettable. What MUI returns when generating a BoopsiObject is a standard MUI object, not a pointer to the BOOPSI gadget itself. In case you really need this BOOPSI gadget pointer, you can obtain it by getting MUIA\_Boopsi\_Object from the MUI object.

Since MUI passes along every unknown attribute to the BOOPSI gadget, there should be no need for this tag anyway.

Note that the BOOPSI object pointer is only valid when the window is open!

## SEE ALSO

MUIA\_Boopsi\_Class, MUIA\_Boopsi\_ClassID

## MUIA\_Boopsi\_Remember

### NAME

MUIA\_Boopsi\_Remember — V4 [I.], ULONG, 0x8042f4bd

### FUNCTION

Most BOOPSI objects are kind of silly, they don't support automatic resizing or jumping from screen to screen. Therefore, MUI sometimes needs to dispose and regenerate a BOOPSI object. This will result in losing the current state of the object, e.g. saturation and hue values in a colorwheel.

To solve this problem, you can tell MUI what attributes must be remembered during dispose/regeneration. For a colorwheel, this would e.g. be WHEEL\_Saturation and WHEEL\_Hue.

Before disposing the BOOPSI object, the remember tags are read and stored in a private buffer. After regeneration, the contents of this buffer are passed back to the BOOPSI object again.

Note that you can define up to five MUIA\_Boopsi\_Remember tags.

### BUGS

The remember procedure will not work when the attributes you want to remember are just pointers to data stored somewhere in the BOOPSI object.

### EXAMPLE

see MUIA\_Boopsi\_ClassID

NAME

## SEE ALSO

MUIA\_Boopsi\_ClassID

## MUIA\_Boopsi\_Smart

### NAME

MUIA\_Boopsi\_Smart — V9 [I.], BOOL, 0x8042b8d7

### FUNCTION

Specify TRUE for smart BOOPSI gadgets that allow resizing, e.g. the textfield.class. In this case, MUI will not dispose and recreate the object.

## MUIA\_Boopsi\_TagDrawInfo

### NAME

MUIA\_Boopsi\_TagDrawInfo — V4 [ISG], ULONG, 0x8042bae7

### FUNCTION

Unfortunately, most BOOPSI gadgets need information on the display environment they will reside in at object creation time. Due to MUI's concept, this information is not available that early.

To solve this problem, MUI doesn't generate the BOOPSI object instantly, creation is delayed until the window containing the gadget is opened.

At this time, MUI fills some values about display environment into the BOOPSI objects creation tag list. You have to tell MUI, what tags are actually needed.

With MUIA\_Boopsi\_TagDrawInfo you can tell MUI where to fill in a needed DrawInfo structure.

### EXAMPLE

If your BOOPSI gadget needs a pointer to a DrawInfo structure supplied with the MYBOOPSI\_DrawInfo tag, you would have to specify

```
BoopsiObject
  RecessedFrame

  MUIA_Boopsi_TagDrawInfo MYBOOPSI_DrawInfo

  MYBOOPSI_DrawInfo

  GA_Left
  GA_Top
```

SEE ALSO

```
GA_Width
GA_Height
End
```

## SEE ALSO

[MUIA\\_Boopsi\\_ClassID](#), [MUIA\\_Boopsi\\_TagScreen](#), [MUIA\\_Boopsi\\_TagWindow](#)

# MUIA\_Boopsi\_TagScreen

## NAME

[MUIA\\_Boopsi\\_TagScreen](#) — V4 [ISG], ULONG, 0x8042bc71

## FUNCTION

Unfortunately, most BOOPSI gadgets need information on the display environment they will reside in at object creation time. Due to MUI's concept, this information is not available that early.

To solve this problem, MUI doesn't generate the BOOPSI object instantly, creation is delayed until the window containing the gadget is opened.

At this time, MUI fills some values about display environment into the BOOPSI objects creation tag list. You have to tell MUI, what tags are actually needed.

With [MUIA\\_Boopsi\\_TagScreen](#) you can tell MUI where to fill in a needed Screen structure.

## EXAMPLE

If your BOOPSI gadget needs a pointer to a Screen structure supplied with the MYBOOPSI\_Screen tag, you would have to specify

```
BoopsiObject
  RecessedFrame

  MUIA_Boopsi_TagScreen MYBOOPSI_Screen

  MYBOOPSI_Screen

  GA_Left
  GA_Top
  GA_Width
  GA_Height
End
```

## SEE ALSO

[MUIA\\_Boopsi\\_ClassID](#), [MUIA\\_Boopsi\\_TagDrawInfo](#), [MUIA\\_Boopsi\\_TagWindow](#)



# MUIA\_Boopsi\_TagWindow

## NAME

MUIA\_Boopsi\_TagWindow — V4 [ISG], ULONG, 0x8042e11d

## FUNCTION

Unfortunately, most BOOPSI gadgets need information on the display environment they will reside in at object creation time. Due to MUI's concept, this information is not available that early.

To solve this problem, MUI doesn't generate the BOOPSI object instantly, creation is delayed until the window containing the gadget is opened.

At this time, MUI fills some values about display environment into the BOOPSI objects creation tag list. You have to tell MUI, what tags are actually needed.

With MUIA\_Boopsi\_TagWindow you can tell MUI where to fill in a needed Window structure.

## EXAMPLE

If your BOOPSI gadget needs a pointer to a Window structure supplied with the MYBOOPSI\_Window tag, you would have to specify

```
BoopsiObject
  RecessedFrame

  MUIA_Boopsi_TagWindow MYBOOPSI_Window

  MYBOOPSI_Window

  GA_Left
  GA_Top
  GA_Width
  GA_Height
End
```

## SEE ALSO

MUIA\_Boopsi\_ClassID, MUIA\_Boopsi\_TagDrawInfo, MUIA\_Boopsi\_TagWindow

---

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

MUI for AmigaOS -  
MUI-Autodocs

Updated: 06-Dec-2018