

SpeedBar-English

COLLABORATORS

	<i>TITLE :</i> SpeedBar-English		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 13, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SpeedBar-English	1
1.1	SpeedBar 14.4	1
1.2	SpeedBar - Introduction	1
1.3	SpeedBar - Author	1
1.4	SpeedBar - Warning, Requirements, Installation and Distribution	2
1.5	SpeedBar - User	2
1.6	SpeedBar - Developer	4

Chapter 1

SpeedBar-English

1.1 SpeedBar 14.4

SpeedBar 14.4
Copyright © 2000–2002 Alfonso Ranieri - Simone Tellini

1.
 - Introduction
 2.
 - Author
 3.
 - WRID
 4.
 - User
 5.
 - Developer

1.2 SpeedBar - Introduction

Introduction

SpeedBar allows you to easily create toolbars similar to those you can see in Windows programs (although you can get a standard look as well ;)

1.3 SpeedBar - Author

Author

SpeedBar was originally created by Simone Tellini <wiz@vapor.com>.

Since version 12.0 the project is maintained and

developed by Alfonso Ranieri <alforan@tin.i>.

The support page is located at:
<http://web.tiscalinet.it/amiga/>

1.4 SpeedBar - Warning, Requirements, Installation and Distribution

Warning, Requirements, Installation and Distribution

Warning

THIS SOFTWARE AND INFORMATION ARE PROVIDED AS IS.
ALL USE IS AT YOUR OWN RISK, AND NO LIABILITY OR
RESPONSIBILITY IS ASSUMED. NO WARRANTIES ARE MADE.

Requirements

The library needs AmigaOS, version 3 or higher.

Installation

Use the installation script.

Distribution

These classes can be freely used: all I ask is to receive
a registered copy of your program if it uses them.

1.5 SpeedBar - User

User

SpeedBar is highly configurable via the MUI
preferences program:

Backgrounds

Group - You may select the background for the whole bar.

To activate it you must select "User" in the cycle
gadget.

Button - You may select the background for a single button.

It will be used if the the bar is RaisingFrame when the
mouse is over the button.

To activate it you must select "User" in the cycle
gadget.

Frame

Shine - You may select the shine pen used to draw the frame
around a button for a BorderLess/RaisingFrame bar,
when the mouse is over the button.

Shadow - You may select the shadow pen used to draw the frame.

Style - If set to Recessed, the above pens are exchanged.

Disabled pens

Body - When a button is disabled, this pen is used to draw

the body of the button.

Shadow - When a button is disabled, this pen is used to draw the shadow of the button.

Bar Spacer

Since V12 SpeedBar offers a special bar as spacer.

Here you may control some behaviours of that.

Shine - The shine pen to draw the bar.

Shadow - The shadow pen to draw the bar.

Use always - Always use the bar instead of pure space as spacer.

Fonts

Text - The font to use for Text-only buttons.

TexGfx - The font to use for Text-and-image buttons.

Settings

Precision - The precision to use when obtaining pens.

Event - The event to use for Mouse-over-button.

Scale factor - The scale factor for Small (scaled) images.

Group spacing

Horiz - The pixels among buttons in a horiz bar.

Vert - The pixels among buttons in a vert bar.

Inner spacing

Horiz - The pixels between the left/right frame and the contents of the button (text and/or image) for a BorderLess/RaisingFrame button.

Vert - The pixels between the top/bottom frame and the contents of the button (text and/or image) for a BorderLess/RaisingFrame button.

Text/Gfx spacing

Horiz - The pixels between the text and the image, if the text is placed at the left or at the right of the image.

Vert - The pixels between the text and the image, if the text is placed at the top or at the bottom of the image.

Note:

1. Since SpeedBar V13, the text may be placed at the top/bottom/left/right of the image.
The programmer decides to let the user modify the text position.
If so, you should consider that choosing a very big font in Fonts/Text/Gfx might result in the window being enabled to open.
2. If you press the right-mouse-button a short popup menu will be shown; its items are:
alfie - set my preferred values
defaults - set default values

1.6 SpeedBar - Developer

Developer

SpeedBar is very easy to use. All you have to do is to supply a description of the buttons and a description of the images to use.

The buttons are defined via MUIA_SpeedBar_Buttons tag. You have to pass a struct MUIS_SpeedBar_Button array,

the fields are:

```

Img      - the index of the image to use for this button
Text     - the label for this button
Help     - an help string to use as ShortHelp
Flags    - flags for this button
Class    - if you subclassed SpeedButton.mcc, pass your
           subclass here.
Object   - this will be filled after the object was created.
           Just on't use it :-)
```

Example:

```

struct MUIS_SpeedBar_Button buttons[] =
{
    {0, "_Get", "Get the disc.", 0, NULL},
    {1, "_Sa_ve", "Save the disc.", 0, NULL},
    {2, "_Stop", "Stop the connection.", 0, NULL},
    {MUIV_SpeedBar_Spacer},
    {3, "_Disc", "Disc page.", 0, NULL},
    {4, "_Matches", "Matches page.", 0, NULL},
    {5, "_Edit", "Edit page.", 0, NULL},
    {MUIV_SpeedBar_End},
};
```

Note that the array must be always be terminated with MUIV_SpeedBar_End.

You may add other button at any time via MUIM_SpeedBar_AddButton and MUIM_SpeedBar_AddSpacer .

The button definition is copied, so you may discard it after use. Help is not copied !

The description of the image may be supplied in different ways:

MUIA_SpeedBar_Images - you load the single images and set up an array of struct MyBrush * . The array may be freed after use (the array, not the BitMaps !).

MUIA_SpeedBar_Pics - you want SpeedBar.mcc to load the single images for you. Pass here a NULL-terminated array of STRPTR, names of the images to load via datatypes. The array me be

freed after use.

MUIA_SpeedBar_Strip - the images are saved as a Strip: all together separated via a single blank pixels column. The strip will be load via datatypes. The number of the buttons is derived via MUIA_SpeedBar_Buttons or as defined in MUIA_SpeedBar_StripButtons (if you want to add buttons later). Pass a STRPTR, the name of the strip, here. It may be freed after use.

MUIA_SpeedBar_StripBrush - as the above but you load the strip by yourself. Pass a struct MyBrush *, definition of the strip, here. It may be freed after use (the pointer not the BitMap!).

When using a strip, all the pics must have the same width. No requester if shown and no error is reported if the supplied strip is invalid. You will see that by yourself :-)

The MUIA_SpeedBar_PicDrawer attribute may be used to define a path for the pics and the strip to be load via datatypes.

Never cache the single buttons objects pointer: if you set the attributes:

- MUIA_SpeedBar_BarSpacer
- MUIA_SpeedBar_Borderless
- MUIA_SpeedBar_ViewMode
- MUIA_SpeedBar_SmallImages
- MUIA_SpeedBar_Sunny

all the buttons are disposed and recreated!

To invoke a method over a buttons use
MUIM_SpeedBar_DoOnButton or MUIM_SpeedBar_GetObject .

Always let the user modify the aspect of the bar via:

- MUIA_SpeedBar_BarSpacer
- MUIA_SpeedBar_Borderless
- MUIA_SpeedBar_ViewMode
- MUIA_SpeedBar_SmallImages
- MUIA_SpeedBar_Sunny
- MUIA_SpeedBar_Raising
- MUIA_SpeedBar_LabelPosition

If possible, put the bar in a Virtgroup: this will not stress the user if very big fonts and or images are used.
