

**RI**

Conversion program

**COLLABORATORS**

	<i>TITLE :</i> RI		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Conversion program	October 9, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>RI</b>	<b>1</b>
1.1	Overview of RI GTMenu Lib V2.5 . . . . .	1
1.2	RI GTMenu Lib V2.5 . . . . .	1
1.3	RI GTMenu Lib V2.5 . . . . .	2
1.4	RI GTMenu Lib V2.5 . . . . .	2
1.5	RI GTMenu Lib V2.5 . . . . .	3
1.6	RI GTMenu Lib V2.5 . . . . .	3
1.7	RI GTMenu Lib V2.5 . . . . .	4
1.8	RI GTMenu Lib V2.5 . . . . .	4
1.9	RI GTMenu Lib V2.5 . . . . .	4
1.10	RI GTMenu Lib V2.5 . . . . .	4
1.11	RI GTMenu Lib V2.5 . . . . .	5
1.12	RI GTMenu Lib V2.5 . . . . .	5
1.13	RI GTMenu Lib V2.5 . . . . .	5
1.14	Example Programs . . . . .	6

---

# Chapter 1

## RI

### 1.1 Overview of RI GTMenus Lib V2.5

#### Overview

This library provides access to the GadTools library functions for setting up and laying out menustrips. Because of its use of Gadtools, this library is for users with OS2 and above.

Users of the current Acid library should have no problems converting to this library, since it has been designed so that the only major differences between them are the slight command name changes. The benefit of using this library is that GadTools takes over all positioning of menuitems, and will automatically take account of things like font sensitivity and items going off screen. You'll also get that nice OS2 feeling to your menuitems, including those lurvly bar items.

If you wish to have proper GadTools menus you need to open your windows with the NewLookMenus flag set. Do this by either specifying NewLookMenus in your taglist for WindowTags, or by adding the value \$200000 to your window flags when you call the Window command.

The library has its own Blitz2 object associated with it: GTMenuList, which looks like:

```
Newtype.GTMenuList
  *MenuList
  *MenuLookup
  numitems.w
End Newtype
```

The latest version of the RED Debugger has this object built in for convenience.

### 1.2 RI GTMenus Lib V2.5

Statement: GTMenuTitle

-----

---

Modes : Amiga/Blitz  
 Syntax: GTMenuItem GTMenuList#,menu,title\$

This creates the specified menutitle in the GTMenuList given. The text to display for the title is given in title\$.

This command fails if 'menu' is greater than 0 but menu title menu-1 hasn't been defined.

### 1.3 RI GTMenus Lib V2.5

Statement: GTMenuItem

---

Modes : Amiga/Blitz  
 Syntax: GTMenuItem GTMenuList#,flags,menu,item[,itemtext\$  
 [,shortcut\$[mutualexclude[,UserData]]]]

This creates the specified menuitem in the GTMenuList. This command works the same as the Acid command MenuItem \*except\* that the flags parameter is different. The flags in this library refer to the flag definitions found in the Amiga Include files.

You'll notice that the itemtext\$ parameter is optional - if you leave this out you'll get a bar item. This bar item will automatically fit perfectly across the menu width and will be unselectable by the user.

This command fails if the specified menu hasn't been defined, or the item is greater than 0 and (item-1) is undefined.

There is now another optional parameter - mutualexclude. This is a bit pattern of menu items that cannot be 'ticked' whilst this item itself is ticked. Use this to have a group of items, out of which only 1 can be selected at one time by a user. The bit pattern applies to items at the current depth into the menulist, e.g. only items on the same menu title can be mutually exclusive. Refer to the Amiga manuals and the example program for this library for more information.

The optional UserData field allows you to associate a longword value with the mmenu item. This longword will be stored with the menuitem when the menustrip is actually created using (CreateMenuStrip). This value can then be retrieved using the command GTGetUserData.

### 1.4 RI GTMenus Lib V2.5

Statement: GTSubItem

---

Modes : Amiga/Blitz  
 Syntax: GTSubItem GTMenuList#,flags,menu,item,subitem[,itemtext\$  
 [,shortcut\$[mutualexclude[,UserData]]]]

Creates the specified subitem in the GTMenuList. Need I say more? The flags are given as defined in the Amiga Include files.

---

You'll notice that the `itemtext$` parameter is optional - if you leave this out you'll get a bar item. This bar item will automatically fit perfectly across the menu width and will be unselectable by the user.

There is now another optional parameter - `mutualexclude`. This is a bit pattern of menu items that cannot be 'ticked' whilst this item itself is ticked. Use this to have a group of items, out of which only 1 can be selected at one time by a user. The bit pattern applies to items at the current depth into the menulist, e.g. only subitems on the same menu item can be mutually exclusive. Refer to the Amiga manuals and the example program for this library for more information.

The optional `UserData` field allows you to associate a longword value with the `mmenu` item. This longword will be stored with the `menuitem` when the `menustrip` is actually created using (`CreateMenuStrip`). This value can then be retrieved using the command `GTGetUserData`.

## 1.5 RI GTMenus Lib V2.5

Statement: `CreateMenuStrip`

---

Modes : Amiga/Blitz

Syntax: `CreateMenuStrip GTMenuList#`

Creates the actual GadTools `menustrip` from a `GTMenuList` object. When you call commands like `GTMenuTitle` the work is performed on an internal list to the library - not an actual menulist. Thus this command translates the internal list into a menulist and performs all GadTools formatting etc.

This command is invoked automatically if it has not been called before you call `GTSetMenu`.

Note also that you must have a currently used screen object to call this command since GadTools needs to know what screen to format your menus for (font sensitivity). Also note that you can call this command over and over again for a given object - thus after adding new items you can call it again to make this item appear in the menulist.

**WARNING:** before calling this command make sure that the `GTMenuList` given is not attached to a window - problems will occur (corrupt menu items) if you call this item whilst the menulist is attached.  
Use `GTUnSetMenu` to remove this menulist from a window.

## 1.6 RI GTMenus Lib V2.5

Statement: `GTSetMenu`

---

Modes : Amiga/Blitz

Syntax: `GTSetMenu GTMenuList#,Window#`

Attaches the specified `GTMenuList` to the specified windows. If the given

---

GMenuList object has not had CreateMenuStrip called on it, CreateMenuStrip is automatically invoked to calculate the menulist.

## 1.7 RI GTMenus Lib V2.5

Statement: GTUnSetMenu

---

Modes : Amiga/Blitz

Syntax: GTUnSetMenu

This command removes the menulist from the current window. You must call this function before you call CreateMenuStrip IF you have already set the menulist to a window (come again? :) ).

## 1.8 RI GTMenus Lib V2.5

Statement: GTMenuChecked

---

Modes : Amiga/Blitz

Syntax: status=GTMenuChecked (GTMenuList#,menu,item[,subitem])

This command is used in conjunction with menuitems that have a check mark. If the item (or subitem) is currently checked (e.g. a tick appears in front of its text in the menu) then this command returns -1, else it returns 0.

NOTE: you must have called either CreateMenuStrip or GTSetMenu before you use this command.

## 1.9 RI GTMenus Lib V2.5

Statement: GTSetMenuChecked

---

Modes : Amiga/Blitz

Syntax: GTSetMenuChecked GTMenuList#,menu,item[,subitem],Off|On

This sets the tick mark on the selected item (or subitem) to the status supplied (on or off). The same rules apply to this command as GTMenuChecked.

## 1.10 RI GTMenus Lib V2.5

Statement: GTFreeMenu

---

Modes : Amiga/Blitz

Syntax: GTFreeMenu GTMenuList#,menu[,item[,subitem]]

This frees the selected menu title/item/subitem from the libraries internal menu list. This change will not be reflected in the actual menulist

---

created by CreateMenuStrip until you recall CreateMenuStrip.

Deleting an item automatically causes its descendants to be deleted, i.e. deleting a menu title will cause all items that belong to it to be deleted, deleting a menu item will cause all subitems that belong to it be deleted.

## 1.11 RI GTMenus Lib V2.5

Statement: GTMenuState

Modes : Amiga/Blitz

Syntax: GTSetMenuState GTMenuList#,menu,item[,subitem],Off|On

This sets the tick mark on the selected item (or subitem) to the status supplied (on=enabled or off=disabled).

## 1.12 RI GTMenus Lib V2.5

Statement: GTGetUserData

Modes : Amiga/Blitz

Syntax: userdata.l=GTGetUserData (GTMenuList#,menu,item[,subitem])

This command can be used to retrieve the userdata associated with a specific menuitem/subitem. The userdata value for a menuitem/subitem is passed as a parameter to the commands GTMenuItem and GTSubItem. The UserData longword can be used to store any value you wish (e.g. could be a pointer to a function, or a pointer to a string etc).

## 1.13 RI GTMenus Lib V2.5

---

| RI GTMenus Lib V2.5 |

---

©1996 Red When Excited Ltd

Undocumented commands added by Toby Zuijdveld 02/03/1999  
mailto: hotcakes@abacus.net.au

Overview

Command Index

CreateMenuStrip

GTFreeMenu

---



