

menu

COLLABORATORS

	<i>TITLE :</i> menu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	menu	1
1.1	GUIEnvironment/Menu guide	1
1.2	Creating a new menu	1
1.3	Localize a menu	3
1.4	The menu help function	3
1.5	Menu message handling	4
1.6	The menu information structure	4
1.7	The menu tags	4
1.8	rcs	5

Chapter 1

menu

1.1 GUIEnvironment/Menu guide

GUIEnvironment

Menu guide

```
=====
© 1994   Carsten Ziegeler
         Augustin-Wibbelt-Str.7
         D-33106 Paderborn
         Germany
=====
```

Creating a new menu

Localize a menu

The menu help function

Menu message handling

The menu information structure

The menu tags

1.2 Creating a new menu

GUIEnvironment uses the gadtools menu functions to create the menu ↔

In form of the CreateGUIMenuEntry function you have an easy to use interface to create own menus.

GUIEnvironment uses the UserData entries of the menu items to store some important information about the menu ! So if you need to your

own user data, use the `userData` entry of the `GUIMenuInfo` structure !

If you want to create the following menu, you have to do the below mentioned calls, after you have created a pointer to a `GUIInfo` structure! The `GUIInfo` structure in this example is named `GUI`.

```

Project      Edit
-----
Load         Cut
Save         Paste
QUIT

CreateGUIMenuEntry(GUI, NM_TITLE, "Project", NULL);
CreateGUIMenuEntry(GUI, NM_ITEM,  "Load",
                    GEM_ShortCut, "L\o", NULL);
CreateGUIMenuEntry(GUI, NM_ITEM,  "Save",
                    GEM_ShortCut, "S\o", NULL);
CreateGUIMenuEntry(GUI, NM_ITEM,  "QUIT",
                    GEM_ShortCut, "Q\o", NULL);
CreateGUIMenuEntry(GUI, NM_TITLE, "Edit", NULL);
CreateGUIMenuEntry(GUI, NM_ITEM,  "Cut",
                    GEM_ShortCut, "C\o", NULL));
CreateGUIMenuEntry(GUI, NM_ITEM,  "Paste",
                    GEM_ShortCut, "V\o", NULL);

```

If you now call `DrawGUI(GUI, NULL)` the menu is attached to the window and now you check the incoming messages.

Using the `GEM_AHook` tag you can specify a hook function which is called every time a `IDCMP_MENUPICK` message arrives:

```

CreateGUIMenuEntry(GUI, NM_ITEM, "Paste",
                    GEM_AHook, &PasteFunction, NULL);

```

The hook function gets in `A0` a pointer to a hook structure. Also the `A4` register is set to the `compilerReg` entry of `GUI`. `A2` is set to the `MenuItem` structure of the menu item and `a1` is currently unused.

```

BOOL __asm PasteFunction(register __a0 struct Hook *hook,
                        register __a2 struct MenuItem *item,
                        register __a1 APTR unused)
{
    /* paste function implementation */
    return(TRUE);
}

```

It is possible to change the menu font later in the program by using `ChangeGUI` and the `GUI_MenuFont` tag. This change will be done immediately.

SEE ALSO

Menu message handling

The `GUIMenuInfo` structure

The menu tags

The menu event hook

1.3 Localize a menu

Localizing the menu is very easy:

First, when creating the GUIInfo structure, you have to specify the name of the catalog file for this GUI and the number of the first menu item text within this catalog.

The catalog should be designed in that way, that the menu item texts are in straight order.

For example: A catalog for the example of the menu creating chapter could look like this:

```
200: Project
201: Load
202: Save
203: QUIT
204: Edit
205: Cut
206: Paste
```

Using the GUI_MenuCatalogOffset tag with the data of 200 and the GUI_CatalogFile tag with the file name of the catalog, you don't need to do anything else. The menu will appear in the correct language (if a catalog exists).

If the Cut and Paste texts would have the numbers 208 and 207 you would have to use the GEM_CatalogString tag for both menu items. For cut you would pass GEM_CatalogString, 208 and for Paste GEM_CatalogString, 207.

With each call to CreateGUIMenuEntry the catalog string number is increased by one. You could check the menuCatalogOffset entry of the GUIInfo structure for the current value.

1.4 The menu help function

If the AmigaGuide is installed and if you have turned on [← intuitions](#)

menu help function, GUIEnvironment displays the node specified in the GUIMenuInfo structure for each item if a IDCMP_MENUHELP message arrives.

Usually CreateGUIMenuEntry creates the node names for you, this means the first item gets the node name MENU0, the second MENU1 and so on.

If you want to have different node names, use the GEM_GuideNode tag.

SEE ALSO

The GUIMenuInfo structure

The menu tags

1.5 Menu message handling

With each IDCMP_MENUPICK and IDCMP_MENUHELP message GUIEnvironment sets the msgMenuNum, msgItemNum and msgSubNum entries of the GUIInfo structure for you. If this message concerns a menu item, also the msgItemAdr entry is set !

With a IDCMP_MENUPICK message, the given hook function for this item is called. If it returns FALSE the application will not get the message, otherwise if you return TRUE the application will hear of the message !

GUIEnvironment also handles the multi selection of menu items !

With every call to GetGUIMsg/WaitGUIMsg you will get the next selection of a multi selection as if they were really selected one by one !

SEE ALSO

Menu help function

1.6 The menu information structure

Each menu item gets the UserData entry set to this structure. This structure is read only. Your own user data is stored in the userData field of this structure !

```
struct GUIMenuInfo
```

```
{
```

```
    APTR userData;
```

Use this for own user data.

```
    struct Hook function;
```

The function which is called with every IDCMP_MenuPick message.

```
    STRPTR guideNode;
```

The AmigaGUIDe node which is displayed with every IDCMP_MenuHelp message

```
};
```

The entries can be changed using the menu tags.

SEE ALSO

The menu tags

1.7 The menu tags

GEM_UserData

The UserData for this menu item.

GEM_AHook

This function is called when an IDCMP_MenuPick message for this gadget arrives.

GEM_GuideNode

AmigaGuide node. This node is displayed with every IDCMP_MenuPick message.

GEM_CatalogString

The number of the menu item text within the catalog.

GEM_ShortCut

A string which contains the key for the short cut.

GEM_Flags

Menu item flags.

GEM_MutualExclude

Mutual exclusion set.

1.8 rcs

\$RCSfile: Menu.guide \$

\$Revision: 1.5 \$

\$Date: 1994/11/03 15:52:15 \$

GUIEnvironment Menu Guide

Copyright © 1994, Carsten Ziegeler

Augustin-Wibbelt-Str.7, 33106 Paderborn, Germany
