

GadUtil.Manual

COLLABORATORS

	<i>TITLE :</i> GadUtil.Manual		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 17, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GadUtil.Manual	1
1.1	GadUtil.library 37.10	1
1.2	Why another GUI creation library ?	2
1.3	System requirements for this library	3
1.4	For those who will copy this around	3
1.5	How to install this package	4
1.6	A brief list the library's features	5
1.7	Limitations (in this version) of GadUtil	7
1.8	history	8
1.9	Some technical info about the library	12
1.10	Known bugs in current release of GadUtil	12
1.11	Fill in form for bug reports	12
1.12	People that in some way helped to make this possible	14
1.13	Contact addresses	15

Chapter 1

GadUtil.Manual

1.1 GadUtil.library 37.10

```

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
|
|          Gadutil.library version 37.10
|
|          Copyright © 1994-96 by Staffan Hämälä and P-O Yliniemi
|
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

```

GadUtil is another gui library which main goal was to make it easy to create font sensitive user interfaces. The library has grown away from this main goal, and become another useful-routine-library.

- Introduction
 - Why another GUI creation library ?
- Requirements
 - System requirements for this library
- Distribution
 - For those who will copy this around
- Installation
 - How to install this package
- Features
 - A brief list the library's features
- Limitations
 - Limitiations (in this version) of GadUtil
- Short history
 - Changes since the first release
- Technical Info
 - Some technical info about the library

- Bugs
 - Known bugs in current release of GadUtil
- Bug report
 - Fill in form for bug reports
- Acknowledgments
 - People that in some way helped to make this possible
- Authors
 - Contact addresses

Links to other docs (only distributed with developer archive)

- Verbose history - Development history (day by day, English)
- Verbose history - Development history (day by day, Swedish)
- Autodocs - GadUtil AutoDocs
- Function Index - GadUtil AutoDocs index

1.2 Why another GUI creation library ?

A little of the history behind this program:

Everybody who have tried to do a program with a nice GUI have almost certainly come across the problem; how to adjust the program based on the font that user has selected. It began as a little test program that should be able to adjust itself, but I soon realized that it wasn't that easy (as I had thought).

I was almost finished with a working version of the program when I got some other programs/libraries that did something similar.

The first program I got was MUI (Magical User Interface). It wasn't that bad, but it was big (170k) and it was slow.

Another program was gadoutline.library, it wasn't exactly like I would want, and it was quite big (60k).

Then, I came across the program that I liked the most: GadLayout 1.5. I changed one of my programs, so that it would use GadLayout, but it didn't work. I think that the problem was that my assembler didn't work with the object file for GadLayout (I had never tried that before).

When GadLayout didn't work for me I continued with my own project, but I changed it a bit, because I liked the way GadLayout used tags. So I changed my program, so that it used tags instead of my own gadget structure. I also tried to make the tags work in a similar way as GadLayout's tags, I also used similar names for the tags (mainly so that I didn't need to change the program I had adjusted to GadLayout so much).

After that the first beta version of the program was ready, a friend of mine, P-O, also started working on the project. We decided to make the program into a shared library, so that all programs could use it without having to include it in the executable file. The library was named GadUtil.

After that moment the library has grown quite a lot, with new tags, new functions, a lot of enhancements and bugfixes. Now the library contains about 47 functions, and has quite a lot of tags. Another feature of the library is that it has some new gadget kinds, like a progress indicator, image gadgets, drawer gadgets etc. It is also very easy to add hotkeys and localization to your programs when using gadutil.library.

/Staffan

1.3 System requirements for this library

- Should run on all Amigas with at least KS 37.175 / OS 2.04 ←
- To take advantage of the locale support OS 2.1 is needed
- diskfont.library makes the font adaptive stuff a bit more reliable

The library is tested on the following systems:

```
A500 KS 3.0 / WB 3.1 / 030-33
A3000 KS 3.1 / WB 3.0 / 030-25
A1200 KS 3.0 / WB 3.0 / 030-50
A4000 KS 3.x / WB 3.x / 030-33
A4000TE KS 3.1 / WB 3.1 / 060-50
```

If there are any problems getting the demo programs, or other programs that uses GadUtil, fill in the Bug report form and send it to peoyli@algonet.se or staham@algonet.se.

1.4 For those who will copy this around

Distribution:

- 1) Distribution of this library is allowed in two ways:
 - 1) Only the .library file(s)
 - 2) The complete distribution (user or developer archive)
 - 2) Programmers of totally free software can use and distribute this library for free.
 - 3) Programmers of Shareware, Licensware, commercial products and other programs that the programmer get any kind of profit from shall send a fully working (and registered) copy of the program and its docs to us. All larger updates of the program should also be sent to us.
 - 4) Programmers of non-free software not covered by 1) and 2) (my Diskware concept belongs to this group) shall be treated as in 3) (a fully working and registered copy...).
 - 5) All programs (except commercial products in some cases) must state
-

that gadutil.library is used. This may be done in either the About requester or in the documentation, but best of all in both places.

Demonstration programs for gadutil.library (that includes the full source) may be distributed without this note.

Example for inclusion in About requester:

```
"This program uses gadutil.library, which
  is Copyright © 1994-1997 by P-O Yliniemi
  and Staffan Hämälä."
```

- 6) All localized programs that uses this library should include the .ct file(s) to allow anyone to translate that program into their own language.

1.5 How to install this package

Depending on which archive you downloaded, you will either have one or three ways to install the complete package, or parts of it on your system.

User archive installation:

Double click on the Install_User icon. This will install gadutil, using the standard installation program by Commodore / Escom. If you don't select "Novice User", you can select which parts to install. You'll only have 4 options here:

```
The library file      - the only file that is necessary to install
The debug version    - useful to trace problems in programs that uses GU
User documentation   - this guide file
Aminet .readme file - the readme files for both user and developer
archive
```

Developer archive installation:

Double click on the Install icon. This will install gadutil, using the standard installation program by Commodore / Escom. If you don't select "Novice User", you can select which parts to install. Select which "kind" of user/programmer you are, and the installation script will fill in some defaults to which parts to install. The next (up to) 10 pages will let you change the default selected parts for the user selection you did earlier.

The installation of the AmigaE related stuff (modules and example programs) is documented in the file "AmigaE/Docs/gadutil_amigae.guide".

Include file installation/removal:

A little shell script is included for fast installation/removal of includes.

This script will also allow you to edit the include files (don't spread

edited includes around you, send them to us if you find any bug).

The script makes use of AskEnv to make it look like a "real" program. ↵
 You
 will probably have to change the editor path/command in the end of the
 InstallInc.gad file before running the script. Double click the ↵
 InstallInc
 gadget to start the program.

Type in the path for your 'C' and/or assembly language include files (if ↵
 you
 use the selector gadget at the left of the string, you may have to edit ↵
 it
 by hand afterwards).

To install includes: select your 'C' compiler or Asm package, and press ↵
 the
 "Install..." gadget.

To remove the includes, you don't have to select the compiler or ↵
 assembler
 type, just press "Remove...".

To edit the includes just press one of the "Edit ... files" gadgets at ↵
 the
 bottom of the window.

1.6 A brief list the library's features

Below is a list of some of the main functions covered by this library.

- * FAST (NO OR VERY SMALL DIFFERENCE AGAINST GADTOOLS) AND SMALL (LESS THAN 13k).
 WRITTEN IN ASSEMBLY LANGUAGE. NO F*CKING 'C' or any other slow language!
 - * Font adaptive. Use any of the system fonts and most other fonts for
 gadgets, texts and menus. All gadgets are placed relative to each
 other, but can be placed at absolute positions (needed for the first
 gadget).
 - * No need to use GIMMEZEROZERO windows, as this is automatically handled
 by the library. If you specify a gadget position of 10,10 the gadget
 will be placed at 10,10 within the window's border.
 - * Localized. It is really easy to localize all your (new) programs by
 using this library. Functions and macros for creating localized menus
 are supplied.
 - * Both assembly and 'C' language support. This is one of the things that is
 missing in most of the other gui libraries. Some extra macros are defined
 for assembly language users, and will make the gadget and menu creating
 process even more easy.
 - * All GadTools gadgets are supported. Some extra functions are provided
 for ListViews. BUTTON_KIND (and IMAGE_KIND) gadgets have some extra tags,
 including a toggle-select option.
-

6 extended gadget types are handled by the library:

- FILE_KIND - gadget for selecting a file.
- DRAWER_KIND - gadget for selecting a drawer. Most programs also use this gadget for file selection.
- IMAGE_KIND - uses an intuition image for the gadget. Can use different images for selected and non-selected state.
- BEVELBOX_KIND - gadtools bevelboxes, with some extensions. All gadtools bevelboxes are supported (even with v37). Extensions allows text to be placed above, in or below the upper line of the box. All kinds can be inverted.

New for v36.52: Bevelbox text placement extended. 3 new placements available together with the old ones: centered, left adjusted & right adjusted.

```
BFT_BUTTON    - GT v37 BUTTON_KIND border
BFT RIDGE     - GT v37 STRING_KIND border
BFT_DROPBOX   - GT v39 BBFT_ICONDROPBOX border
BFT_HORIZBAR  - A horizontal shadowed line
BFT_VERTBAR   - A vertical shadowed line
```

This makes it a total of 100 bevel box types. The bevel boxes are treated as gadgets by the library, and you can obtain the structure for the bevelbox in the same way as you get the gadget pointer.

New for v36.53: Added optional shadow on bevel box text. The shadow has a selectable color/pen and the "sun" can be placed in four different positions.

- PROGRESS_KIND - progress indicator gadget. Background and fill color can be selected. Supports values up to 4.294.967.295, but maximum value should not exceed 4.294.967.295/width. v36.55.
- LABEL_KIND - a text label that can be placed relative to other texts, gadgets, bevelboxes and progress indicators.

- * Can create the minimum required IDCMP mask so that all gadgets works.
- * A new OpenFont (graphics.library and diskfont.library replacement) is supplied. This function will open any font (from RAM, ROM or load it from disk).
- * Functions for listviews include adding nodes (memory for the string is allocated by the function), counting nodes, getting node pointers, clearing a listview, attach and detach lists, moving nodes up and down. A function that allows you to sort one or two lists is included from v36.59.
- * No need to open gadtools.library anymore, since all gadtools functions are provided through this library. Some (the most useable) functions from locale library are also provided by gadutil.

- * Many libraries are opened through gadutil.library, and these library bases are available for your own use (you don't have to open or close some libs).
- * All gadgets are created using tag lists. Every next gadget will use the previous gadgets values as defaults. Only one call to the gadget making routine is needed for every gadget list.
- * All gadgets can have their own hotkey. No extra work is needed by the programmer (only one extra tag).
- * All gadgets (and bevelboxes, texts and progress displays) can have a help string attached to it. If the pointer is moved above that object, a selectable string or text gadget will be used to display the help string. The text can alternatively be shown in the window's or screen's title. The help string can be localized.
- * Distributed with lots of examples, mostly in assembly language, but also in 'C'.
- * Other functions include:
 - Status change for toggle select buttons.
 - String change for STRING and TEXT_KIND gadgets.
 - Disable functions for all gadget kinds (new for 36.56).
 - Routines to set up a busy-pointer (available for both v37 and v39). These routines also blocks the input to the main window of the program.
 - Routine to check if the mouse pointer is within a gadgets box. Useful for mouse-position sensitive appwindows.
 - Bevelbox and window refreshing functions.
 - Functions to clear and size/move windows
 - Easy access to the most important values of the gadgets.
- * FAST (NO OR VERY SMALL DIFFERENCE AGAINST GADTOOLS) AND SMALL (LESS THAN 13k). WRITTEN IN ASSEMBLY LANGUAGE. NO F*CKING 'C' or any other slow language!

1.7 Limitations (in this version) of GadUtil

The following limitations exist in the current version of GadUtil:

- Gadgets will not be recalculated when resizing a window
 - Some GadgetID's are reserved by GadUtil. This should not be a problem, since at least 65000 GadgetID's will still be usable by the programmer. Currently there are only 3 reserved ID's (65533 to 65535, or -1 to -3).
-

1.8 history

Changes overview:

* 37.10 (28-Sep-97)

GU_NoCreate now works on individual gadgets. Useful to create invisible gadgets that helps you place the other objects in the window.

6 new tags added that will allow you to save the object's calculated left, top, right and bottom edge, as well as the gadget's width and height.

E includes added to the distribution.

* 37.9 (22-Dec-96)

6 new tags added for the help functions. It is now possible to set a default help text that will be shown if no gadget is under the pointer.

GU_DefWTitle	- default window title
GU_DefLocWTitle	- localized window title
GU_DefSTitle	- default screen title
GU_DefLocSTitle	- localized screen title
GU_DefHelpText	- default help text
GU_DefLocHelpText	- localized help text

A little bug removed from the help routines. Versions before 37.9 could display two different help text at the same time (in different gadgets or the screen or window title).

A new include file is included in the distribution now. Now it should be possible to compile the examples with older versions of the system include files (OS 2.0x includes are supported now)

A third archive with the HTML version of the GadUtil documentation is included in the distribution now. Also included with this archive is some pictures of applications using GadUtil.

The distribution has changed a bit, resulting in that the archive with all files became smaller (developer archive only).

* 37.8 (15-Jul-96)

Some bug fixes in the new help functions. If 0 was used as data on a GU_HelpText tag with a gadget that shows the help text in the window's or screen's title, there were some enforcer hits

Deleted the separate autodoc files. Appended some information on the contents of the GU_Public structure to the autodocs. More info about SCRTITLE_HELP and WINTITLE_HELP was also appended.

Changed the help function so that all gadgets and title bars that are used to

display help strings is cleared if the mouse pointer is outside all ↔
gadgets that
have help texts.

* 37.7 (12-Jul-96)

Bug fix in the prototype for GU_ChangeStr (two parameters was mixed up).

GU_ChangeStr now works for both STRING_KIND and TEXT_KIND gadgets. If the gadget isn't a STRING or TEXT gadget nothing is done.

Bug fix in GU_CoordsInGadBox function. The register D3 was trashed in all earlier versions of GU.

Three new tags allows any STRING or TEXT_KIND gadget to show a (localized) help text for each gadget in the window. These tags also works with the special gadgets (BEVELBOX, LABEL, PROGRESS). The gadgets attached to the window is scanned in sequential order. If a BEVELBOX_KIND gadget matches the pointer position, this gadget's help string will have the lowest priority. If another BEVELBOX is found inside this, the help string for that one will be used if there is no gadget inside that box. The help text can also be displayed in the window/screen title.

The TEXT_KIND, STRING_KIND, NUMBER_KIND and INTEGER_KIND gadget's current value can now be read from the gu_Active field in the structure that gg_UserData points to.

* 37.6 (15-Jun-96)

gu_Active in the structure that gg_UserData points to was not updated when using any routine that used GU_SetGadgetAttrs(). Now it checks the gu_Active value to see if it is out of the gadget's limits.

gng_Flags and gng_GadgetText is now cleared before creation of every new gadget.

The LAYOUTA_Spacing tag for LISTVIEW_KIND gadgets is now used to determine the height of the gadget. If the height of a gadget should be about 4 lines (GU_HeightFactor, 4), the height of the gadget is calculated to 4*(font height+value of LAYOUTA_Spacing tag). The same method is used for the GU_AddHeiLines tag.

Hotkeys for LISTVIEW_KIND gadgets works as they should again. In v37.5 of GadUtil (and probably some earlier ones), the listview didn't scroll down when you selected the next entry.

Inserted pragmas and prototypes for tagcalls (pragmas only for SAS/C). It is possible to create stub routines for the tagcalls for other compilers by using fd2pragma 2.0.

Inserted links to the include files in the index file for GadUtil.guide.

* 37.5 (29-May-96)

Bug in PROGRESS_KIND gadget removed. If the gadget was set to show x/0 (also 0/0), a division by zero was done.

Two new constants added to the include files: GADUTIL_VER and GADUTIL_REV.

Bug fixes in test (demo) programs. The FPF_DISKFONT flag in the TextAttr structure was unset in all demo programs. Replaced the manual version check in the demo programs with the GadUtil function GU_CheckVersion. Removed unnecessary IDCMP flags from the demo programs and replaced GadTools IDCMP with the GU_MinimumIDCMP tag.

* 37.4 (14-May-96)

Created a new gadget kind: LABEL_KIND. This gadget kind is used to place text relative to other gadgets, bevel boxes, and other texts. Can use the same type of shadows that the text for the bevel box.

Implemented a new function in the library: GU_CloseFont. Works like graphics CloseFont, but takes the parameter in A0 instead of A1.

Replaced the list sort routine with a new one that sorts better.

Wrote a new document, "GadUtil.index", that have a list over all gadutil functions, sorted by what they do. There are also some shortcuts to all tags for gadutil's own gadget kinds.

Created two new flags for BEVELBOX_KIND and LABEL_KIND:
BB_3DTEXT - can be used as a replacement for GUBB_3DText, TRUE
LB_3DTEXT - can be used as a replacement for GULB_3DText, TRUE

* 37.3 (10-May-96)

Enforcer hits fixed in the code that handles the GU_AdjustLeft and GU_AdjustRight tags.

A new function to clear the inner part of a window was added.

A new function to size (and move, if necessary) a window was added.

Bug fix in assembly language include file.

* 37.2 (02-May-96)

Started programming in StormC and did some tests with the library, noticed that there was a couple of errors in some include files.

GU_UpdateProgress was wrongly specified in the FD/clib files. The output of GU_GetGadgetPtr was specified as (struct Gadget *), this is now corrected.

Included an example in C that uses the PROGRESS_KIND gadget.

* 37.1 (01-May-96)

IMAGE_KIND gadget bug fix. The color of the unselected image of the gadget was wrong, if the image used more than 4 colors.

* 37.0 (29-Apr-96)

A new function that checks the library's revision added. Two new tags added: GU_FractWidth and GU_FractHeight. Suggested by Morten Amundsen.

Some small bug fixes in 'C' include files (unterminated comments).

Cleaned up BetterTest.c a bit (removed some {}). Bug fix on the last line of the example (GT_ReplyIMsg -> GU_ReplyIMsg).

Added support for StormC in both installer script and include-file installer. Created dictionary for StormC, so that all words that belongs to GadUtil (functions, tags and constants) is shown in another color.

- * 36.60 (24-Feb-96)
Uses the screen's font as the gadget font if no GU_DefTextAttr or GU_TextAttr tag is used. Bug fix from v36.59. Changed some docs, assembled all examples, changed installation scripts.
 - * 36.59 (06-Jan-96)
More docs completed (all docs, finally). Bug fix in the GU_AutoWidth, GU_AdjustLeft and GU_AdjustRight tags. The underscore character was counted as a normal one. This made gadgets without hotkeys one char smaller than other gadgets. A new function that sorts one or two lists was added. Some new constants added in include files. Changed GU_MinimumIDCMP to support hotkeys. Changed PROGRESS_KIND to use GU_GadgetText and GU_LocaleText. Uses the screen's font as the gadget font if no GU_DefTextAttr or GU_TextAttr tag is used.
 - * 36.58 (01-Jan-96)
SLIDER_KIND and SCROLLER_KIND gadgets returned wrong result in im->Code if the gadget was changes using the hotkey. Bug reported by Lukasz Szlag.
 - * 36.57 (10-Dec-95)
Some more docs finished. A memory loss bug was fixed.
 - * 36.56 (10-Oct-95)
Changed some routines, so that hotkeys can't be used on disabled gadgets. Made it possible to disable IMAGE_KIND, FILE_KIND and DRAWER_KIND gadgets.
 - * 36.55 (30-Sep-95)
Bug fix in the hotkey part of the library. Wrong code were sent back when using GU_LabelHotkey with a LISTVIEW_KIND gadget. A new gadget kind was created, PROGRESS_KIND.
 - * 36.54 (06-Jul-95)
Added support for GU_AutoWidth to CYCLE_KIND gadgets.
 - * 36.53 (05-Jul-95)
3 new tags and 4 new flags was added for the BEVELBOX_KIND gadget. BEVELBOX_KIND gadgets can now have a shadow added to the text in the border. The "sun" can be placed in 4 different places. GU_MinimumIDCMP tag was added.
 - * 36.52 (25-Jun-95)
First public release. Some new flags for the BEVELBOX_KIND gadget was added. Supports a total of 100 bevelbox kinds.
-

1.9 Some technical info about the library

```

Assembly language code lines:      17.650 (511k)
'C' language code lines:          2.350 ( 72k)
Documentation & scripts:          9.250 (435k)
Development time (days):         66 + undocumented days
Library size:                     12.5k
Number of functions (Dec-94):      11
Number of functions (Jun-95):     46 (first release)
Number of functions (now):        52
Number of tags:                   88
Number of constants:              69

```

These numbers do not include some machine generated code (GadUtil.Guide) or the include files used by the library. The most of all lines is written from scratch.

1.10 Known bugs in current release of GadUtil

No known bugs. See
 Limitations
 for some points that probably will be
 considered as "bugs" by someone. If you find a bug, please fill in the
 Bug report form
 and send it to either me (peoyli@algonet.se) or Staffan
 (staham@algonet.se).

1.11 Fill in form for bug reports

System information:

```

[ ] A500      [ ] A600      [ ] A1200
[ ] A2000    [ ] A3000    [ ] A4000

```

Kickstart version: _____ Workbench version: _____ ↵

CPU type/speed: _____ FPU type/speed: _____ ↵

Version of gadutil.library: _____

Patches / commodities:

```

[ ] CycleToMenu _____ [ ] Yak _____ ↵
_____

```

```

[ ] MultiCX _____ [ ] MCP _____ ↵
_____

```

[] KCommodity _____.

Other: ↔

_____ ↔

_____ ↔

_____ ↔

_____ ↔

_____ ↔

Memory / HD:

_____ Mb CHIP RAM _____ Mb FAST RAM _____ ↔
 Mb HD

Program name/version:

_____ ↔

Bug description:

_____ ↔

_____ ↔

_____ ↔

_____ ↔

_____ ↔

Steps to make the bug appear:

_____ ↔

_____ ↔

Special thanx must go to:

- Amiga Inc. / Commodore / Escom / Amiga Technologies / Viscorp
 - For making everything possible
- Rune Gram-Madsen
 - For the best integrated assembly language development package
- Price (TFA)
 - For the continued development of Asm-One (since 1991). The complete distribution of Asm-One 1.29 is available from both mine and Staffan's home page, see
Authors
for addresses.
- Timothy J. Aston
 - For the inspiration, GadLayout 1.5 (a link library for 'C' programmers)
- Bengt Giger, Jochen Wiedmann, Pete Goodeve
 - AskEnv, FD2Pragma & FlexCat, XIcon
- Nico Max
 - For "forcing" us to release gadutil a bit sooner than it was meant to.
- Lukasz Szelag
 - For the first, and the three following bug reports
 - For ideas about some new constants for the include files
- Morten Amundsen
 - For the suggestion of two new tags (so far)
- Michael Ranner
 - AmigaGuide2HTML (although I had to do some manual editing afterwards)
- James McOrmond
 - For some suggestions and reporting that the examples didn't compile using the OS 2.x system includes :)
- Terje Pedersen
 - For the E include files.
- Mika Lundell
 - For the suggestion about making GU_NoCreate work for individual gadgets, and for the idea to the new tags that can save all calculated values in external variables.

1.13 Contact addresses

If you want to send comments, bug reports, suggestions or anything else, use one of the following addresses:

Per-Olof Yliniemi
Aspvägen 4E
SE-957 32 Övertorneå

Staffan Hämälä
Vegagatan 2 3TR
SE-172 34 Sundbyberg

SWEDEN

e-mail:

peoyli@algonet.se

WWW:

<http://www.algonet.se/~peoyli>

phone:

+46-(0)927-12293

SWEDEN

e-mail:

staham@algonet.se

WWW:

<http://www.algonet.se/~staham>

All e-mail (that have something to do with GadUtil) will be answered.

The latest version of GadUtil is always available for download from our homepages. The best assembly language development system, Asm-One 1.29, can also be downloaded from both mine and Staffans homepage.

Asm-One support page: <http://www.algonet.se/~peoyli/asm-one>
