GadToolBox

	COLLABORATORS		
	TITLE : GadToolBox		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 6, 2023	

	REVISION HISTORY		
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

GadToolBox

1.1 GadToolsBox Documentation

Welcome to the GadToolsBox v2.0b User Manual.

Please select any of the topics listed below and follow up on the links as
you wish.

```
@{ " Disclaimer               " link disclaimer }
@{ " Copyright and Distribution " link copyright }
@{ " Updates                  " link updates }
@{ " Introduction             " link introduction }
@{ " Requirements             " link requirements }
@{ " *Read This*              " link readthis }
@{ " Starting GadToolsBox     " link starting }
@{ " Making a Gadget          " link making }
@{ " Gadget Selection         " link selection }
@{ " Moving and Sizing Gadgets " link movesize }
@{ " The Program Menus        " link menus }
@{ " Gadget Requesters        " link requesters }
@{ " The Menu Editor          " link menueditor }
@{ " The Generated Source     " link source }
@{ " The IDCMP Handler        " link handler }
@{ " The Keyboard & Mouse     " link keyboard }
@{ " The GetFile BOOPSI Image " link boopsi }
@{ " Font Adaptivity          " link adaptivity }
@{ " Credits                  " link credits }
@{ " Notes                    " link notes }
```

1.2 Disclaimer

The author cannot be held liable for the suitability or accuracy of this
manual and/or the program(s) it describes. Any damage directly or
indirectly caused by the use or misuse of this manual and/or the program
it describes is the sole responsibility of the user her/him self.

1.3 Copyright and Distribution

```
                  GadToolsBox release 2.0b


        (C) Copyright 1992-93 Jaba Development
            Written using DICE C v2.07.54R by
                   Jan van den Baard
                   Bakkerstraat  176
                   3082 HE Rotterdam
                       Holland


      Fido: 2:285/502.7 (Jan van.den.Baard)
          UUCP: jaba@desert.wlink.nl


           Oberon Source Generator by
                Kai Bolay (AMOK)
               Hoffmannstraße 168
                D-7250 Leonberg
                    Germany


           Fido: 2:2407/106.3
         UUCP: kai@amokle.stgt.sub.org


                  User Manual
      (C) Copyright  1991-93 Jaba Development


    This program uses both the powerpacker.library
        and reqtools.library  (OS 2.0 version)
    These two libraries are  (c) Copyright Nico François
                        __ __
                       / / /  \
              __      / / / /\ \
        _____\ \__/ /_/ /_/ /_____
             (_____)
```
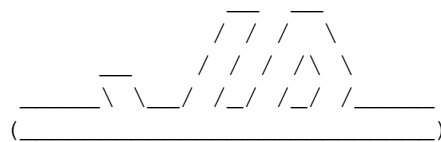
GadToolsBox, (C) Copyright 1991-93 Jaba Development. All rights reserved.
This program is GIFTWARE which means that if you like the program you
should reward the author with gift that, you think, matches his efforts.

This program may be freely distributed as long as all files are included
in the distribution without any modifications.   You may *NOT* charge more
than Fred Fish does for a single library disk.

This program may *not* be uploaded on BBS's that claim copyrights on the
uploaded material.

When you use this program to design GUI(s) for either a commercial or a
shareware program I would appreciate a little note about this in the
program it's documentation.

Commercial distribution of GadToolsBox and/or it's relative files is *NOT*
allowed without written permission from the author.   This also means the
distribution on cover disks, disk magazines etc.

I do not know how this is in other countries but here in holland and in
germany some PD distributors think that they can charge extra if they

translate the manual in their language and distribute that with the
program.

THIS IS NOT ALLOWED! You are welcome to add a translation to this manual
in the archive but you may NOT CHARGE EXTRA FOR THIS!

The source and binary output of GadToolsBox do not fall under the above
mentioned rules. They fall under the rules that the person(s) who created
these files make.

1.4 Updates

Updates are *not* available through me anymore. I have had a small update
service which simply took to much time to keep up. For updates you are
dependand on me releasing newer versions. Please do not send me update
requests anymore.

1.5 Introduction

GadToolsBox is a program that will save you a gigantic amount of time and
anguish in creating a GadTools user interface for your programs. This
program is a follow-up of my other source generator "PowerSource". It's
main purpose is to let you create your user interface without having to
type in the necessary code and structures yourself. This version of
GadToolsBox has the following features:

    -: A complete intuition/gadtools user interface
    -: Editing on many different screen resolutions
    -: Editing/Generating source on/for big autoscrolling screens
    -: All GadTools gadget kinds supported + a custom gadget
    -: GadTools menus supported
    -: Full control over the window flags and most tags
    -: Editing IntuiTexts for the window
    -: Editing, loading and saving of colors (ILBM)
    -: Fully mouse and keyboard controlled
    -: Generation of complete routines in either C, Assembly or Oberon
    -: Generation of "RAW" Assembly source (No headers necessary)
    -: Generation of "static" data
    -: Binary saving for later loading and editing
    -: The possibility to save powerpacker __CRUNCHED!__ binaries.
    -: The ability to print a status report.
    -: Multiple windows
    -: User selectable font
    -: The editing of "DRIPENS" (what gives OS 2.0 that special look)
    -: The possibility to change screen resolution while editing
    -: The possibility to edit "BevelBoxes"
    -: Editing in a user-definable grid
    -: The editing of minimal window border offsets
    -: Easy gadget alignment and spacing
    -: The generation of font-adaptable code (Thanks Sebastiano...)
    -: Multiple gadget moves, deletes, edits and copies.
    -: The generation of event-handlers.

```
-:  etc. etc. etc........
```

1.6 Requirements

GadToolsBox was programmed with DICE C v2.07.54R (Thanks Matt) on a A2500
Kickstart V37.175 Workbench V37.67 with 7 MB memory + ECS + A2630
accelerator card (50MHz!!) + 52 MB harddisk. It should however run happily
on a 1 MB machine without a harddisk and special chips but at least
1.5 MB and a harddisk is suggested.

You need at least Kickstart V37++ and Workbench V37++ to run GadToolsBox.
A C Compiler (try registering DICE) and/or an assembler with the 2.0
headers can also come in handy when you try to compile/assemble the
generated source. Note however that the program can generate assembler
source which does not require the headers to assemble.

The program needs the following libraries:

```
reqtools.library          version 38 or better (required)
icon.library              version 36 or better (required)
nofrag.library            version 2  or better (required)
diskfont.library          version 36 or better (required)
iffparse.library          version 37 or better (required)
gadtoolsbox.library       version 38 or better (required)
powerpacker.library       version 35 or better (optional)
```

If you have the nofrag.library v2.2 please use it instead of v2.1. Version
2.2 is identical in function except that it has some bugs removed. If you
also install the powerpacker.library v35++ you are able to load and/or
save crunched binary files.

1.7 *Read This*

There are older versions of the powerpacker.library V35 that cause a GURU
when decrunching data.   This GURU occures when you have both enforcer and
mungwall running at the time the library is decrunching data.

1.8 Starting GadToolsBox

You can run GadToolsBox from either the Shell or the Workbench. To start
GadToolsBox from the Shell you must type the following:

```
GadToolsBox [name]
```

Where "name" is an optional name of a previously created file which you
want to edit. To start GadToolsBox from the Workbench you should either
double-click it's icon or double-click the icon of a previously created
file.

After a successfull start you are presented with a screen on which a

little window is displayed. This window is the "drawing board" on which
you design your GUI.

When you started GadToolsBox with an argument the contents of the window
edited last will be displayed.

1.9 Making a Gadget

Making a gadget is very simple. First of all you have to select the kind
of gadget you wish to make. To do this you should hold down the right
mouse button and go to the Kind item under the Gadgets menu.

Now you see a list of possible gadget kinds you can choose from.   Select
the kind you want and release the right mouse button.

Now mouse the mouse pointer to where you want the top-left corner of
the gadget to be.

Click on the left-mouse button and move the mouse while holding down
the left mouse button. You see a box which is following your movements.
Now move the box to the exact size you want it to be and release the
left mouse button.

Now a requester pops up in which you can edit the gadget kind it's
attributes.   Please refer to the @{ " Gadget Requesters " link requesters } ←↩
   chapter for more
information on this subject. For now just click on the OK gadget and you
are done. You have created your first gadget with GadToolsBox.

Note that the @{ " GETFILE " link getfile }, @{ " CHECKBOX " link checkbox }   ←↩
   and @{ " MX " link mx } gadgets have a fixed size
which means that you cannot make them any size you wish.

1.10 Gadget Selection

To select a gadget you only have to click on it with the left mouse button.

To select multiple gadgets a method simular to the Workbench icon selection
is used. While you click on the gadgets you want to select you must hold
down a SHIFT key.

It is also possible to select multiple gadget by dragging a box around
them. This method is simular to multi-selecting icons on the Workbench
except that you have to press SHIFT to get into DragSelection mode.

This is how it works:

    a) Press and hold down a SHIFT key.
    b) Click on the left mouse button anywhere in the window where there is
       no gadget located. You can release the SHIFT key now.
    c) Drag the box so that the gadgets that you want to select are partly
       or completely covered by the box.

    d) Release the left mouse button and you are done.

You can see when a gadget is selected when it has 8 little dots surrouding
it. Please refer to the @{ " Moving and Sizing " link movesize } chapter for ←
    information on the
usage of these dots.

1.11   Moving and Sizing Gadgets

Moving and sizing gadgets is really simple. You can choose to move or size
a single gadget or multiple gadgets at the same time.

To move one gadget you must click on the gadget using the left mouse
button. Now while hoding down the left mouse button you should move the
box to it's new location. Now release the mouse button and you are done.

To size a single gadget you click on the gadget using the left mouse
button. ..... It's OK, you can release the mouse button now. Now you see
eight small dots surrounding the gadget (somewhat simular to what you see
here:)

```
  1        2       3
   \       |      /
    o-----o-----o
    |           |
    |           |
4--o           o--5    <--- My pathetic attempt to a visual representation
    |           |               of what you actually see....
    |           |
    o-----o-----o
   /       |      \
  6        7       8
```
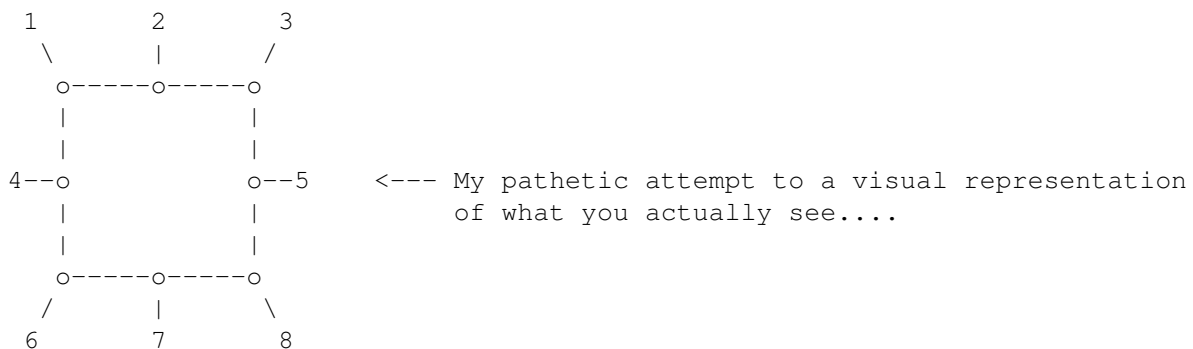
These dots are the points which are used to size the gadget. This is how
they operate:

    1) Size the gadget in the north-west direction.
    2) Size the gadget in the north direction.
    3) Size the gadget in the north-east direction.
    4) Size the gadget in the west direction.
    5) Size the gadget in the east direction.
    6) Size the gadget in the south-west direction.
    7) Size the gadget in the south direction.
    8) Size the gadget in the south-east direction.

For example if you wish to size a gadget in the east direction you select
the gadget. Click on dot number 5 and move the mouse right while holding
down the mouse button. When you sized it enough you release the mouse
button again.

Moving and sizing multiple gadget works simular to moving and sizing a
single gadget.

1.12   The Program Menus

```
This chapter describes the function of each GadToolsBox  menu.   Following
is a list of the menus of GadToolsBox:

Project
    @{ " New...          " link new }
    @{ " Open...         " link open }
    @{ " Save            " link save }
    @{ " Save As...      " link saveas }
     Generate Source
        @{ " C...          " link csource }
        @{ " Assembler... " link asmsource }
        @{ " Oberon...    " link obersource }
     Preferences
        @{ " Main...        " link mainprefs }
        @{ " C Source...   " link cprefs }
        @{ " Asm Source... " link asmprefs }
    @{ " Close Workbench " link closewb }
    @{ " About...        " link about }
    @{ " Quit...         " link quit }

Gadgets
    Kind
        @{ " GETFILE       " link getfile }
        @{ " BUTTON        " link button }
        @{ " CHECKBOX      " link checkbox }
        @{ " INTEGER       " link integer }
        @{ " LISTVIEW      " link listview }
        @{ " MX            " link mx }
        @{ " NUMBER        " link number }
        @{ " CYCLE         " link cycle }
        @{ " PALETTE       " link palette }
        @{ " SCROLLER      " link scroller }
        @{ " SLIDER        " link slider }
        @{ " STRING        " link string }
        @{ " TEXT          " link text }
    @{ " Gadgets         " link gadgetsmenu }
    @{ " Copy            " link copy }
    @{ " Delete...       " link delete }
    @{ " Edit...         " link edit }
    @{ " Join            " link join }
    @{ " Split           " link split }
     Align
        @{ " Left           " link left }
        @{ " Right          " link right }
        @{ " Top            " link top }
        @{ " Bottom         " link bottom }
     Clone
        @{ " Width          " link xsize }
        @{ " Height         " link ysize }
        @{ " Width & Height " link xysize }
     Spacing
        @{ " Set X...       " link xspace }
        @{ " Set Y...       " link yspace }
     Spread
```

```
        @{ " Horizontally   " link xspread }
        @{ " Vertically     " link yspread }
     Transform
        @{ " MX->CYCLE      " link mxcycle }
        @{ " CYCLE->MX      " link cyclemx }
    @{ " Edit TabCycle Order... " link tabcycle }
    @{ " Load...              " link loadgadgets }
    @{ " Save Selected...     " link selectedsave }
    @{ " Save...              " link savegadgets }

Window
    @{ " New...         " link newwindow }
    @{ " Delete...      " link deletewindow }
    @{ " Other...       " link otherwindow }
    @{ " Edit Data...   " link datawindow }
    @{ " Print Info...  " link print }
    @{ " Load...        " link loadwindow }
    @{ " Save...        " link savewindow }
    @{ " Edit Flags...  " link windowflags }
    @{ " Edit IDCMP...  " link idcmp }
    @{ " Edit Tags...   " link windowtags }
    @{ " Edit Grid...   " link grid }
    @{ " Edit Offsets... " link offsets }
     Texts
        @{ " Add...       " link addtext }
        @{ " Edit...      " link edittext }
        @{ " Delete...    " link deletetext }
        @{ " Move...      " link movetext }
        @{ " Load...      " link loadtext }
        @{ " Save...      " link savetext }
     BevelBoxes
        @{ " Add         " link addbox }
        @{ " Move        " link movebox }
        @{ " Size        " link sizebox }
        @{ " Delete      " link deletebox }
        @{ " Flip Recessed " link recessed }
        @{ " Flip DropBox  " link dropbox }
        @{ " Load...      " link loadbox }
        @{ " Save...      " link savebox }

Screen
    @{ " Palette...      " link screenpalette }
    @{ " Select Font...  " link getfont }
    @{ " Edit DriPens... " link dripens }
    @{ " Edit Tags...    " link screentags }
    @{ " Change Type...  " link screentype }

Menus
    @{ " Edit...         " link menueditor }
    @{ " Test            " link testmenu }
    @{ " Load...         " link loadmenu }
    @{ " Save...         " link savemenu }
```

1.13 Project/New...

This will reset the program to it's default settings and erase all windows currently in memory. When you have made chages that are not saved yet the program will inform you about this and it will offer you a safe way out.

1.14   Project/Open...

This will present you with a filerequester in which you must select the GUI that you wish to load. When you have made changes to the GUI in memory that have not been saved yet you will be asked if you want:

```
A)  Save First  -   First save the GUI currently in memory.
B)  Continue    -   Continue with the opening of a new GUI.
C)  Cancel      -   Abort the operation.
```

When the loading operation is finished you will see the window that has been edited the last time you saved the GUI.

1.15   Project/Save

This will save the GUI currently in memory to the name under which it was loaded. When it is a newly created GUI the name used to save it to will be "unnamed.gui".

1.16   Project/Save As...

This will present you with a filerequester in which you must select the name which will be used to save the GUI.   Note that the name will always get the suffix ".gui" appended to it.

1.17   Project/Generate Source/C...

This will present you with a filerequester in which you must select the name which will be used to save the generated C source. Please refer to @{ " The Generated Source " link source } chapter for more information.

1.18   Project/Generate Source/Assembler...

This will present you with a filerequester in which you must select the name which will be used to save the generated Assembler source. Please refer to @{ " The Generated Source " link source } chapter for more information.

1.19   Project/Generate Source/Oberon...

```
This will present you with a filerequester  in which  you must  select the
name which will be used to save the generated Oberon source.  Please refer
to @{ " The Generated Source " link source } chapter for more information.
```

1.20   Project/Preferences/Main...

```
With the main preferences  editor  you  can  control  options  that  will
affect the course of the program.   The  following gadget are in the main
preferences window:
```

```
    User Name           In this gadget you can  type  in  your  own  name.
                        Currently this name will only be included into the
                        generated source.

    Icon Path           In this gadget you should type the  path  in which
                        the GadToolsBox executble can be found.   This path
                        will be used to make a default tool for  the icons
                        of the saved GUI(s).   The  gadget to the  right of
                        the string gadget can be used  to pop  up a  file-
                        requester in which you can select the path.

    Crunch              This gadget allow you to turn on or off the option
                        to   let GadToolsBox crunch  your  saved  files.
                        When this option is on you  can  save upto  85% of
                        disk space.

    Password            Turn on or off the protected crunching.  When this
                        option is on the program will prompt you  to enter
                        a password before it saves the file. You will need
                        this password when you try to load the file again.
                        NOTE:   This option  only  is  available  when the
                                Crunch option is on.

    Buffer              This gadget controls the buffer size which is used
                        to crunch the data.  You can  choose  from  Small,
                        Medium and Large.   The biggest buffer will crunch
                        fast but it takes up a lot of memory.
                        NOTE:   This option  only  is  available  when the
                                Crunch option is on.

    Type                This controls the crunch type used. You can choose
                        from Fast,  Mediocre,  Good,  Very Good  and Best.
                        Using Fast will cruch very  fast but the resulting
                        file will not be crunched as far as it  could  be.
                        Using Best will  crunch  somewhat  slower  but the
                        resulting file will be crunched the best way.
                        NOTE:   This option  only  is  available  when the
                                Crunch option is on.

    Coordinates         This allows you to turn the coordinates on or off.
                        The   coordinates  constantly  display  the  mouse
                        position relative to the upper left hand corner of
```

                              the window.

    Write Icon   Usually GadToolsBox writes an icon to accompany
                       the saved GUI so that you can use it from the
                       Workbench. If you are a shell freak or you simply
                       do not want these icons you can tell GadToolsBox
                       not to save an icon by turning this gadget off.

    GZZ Adjust   GadToolsBox allows you to edit your GUI on a
                       GIMMEZEROZERO window. Normally when you switch
                       from editing on a normal window to a GIMMEZEROZERO
                       window or vice versa all gadgets, texts   and
                       borders will move. This is because on normal
                       windows the coordinates are relative to the top-
                       left of the window and on GIMMEZEROZERO windows
                       they are relative to the top-left of the window
                       inside the window border area. When this option
                       is turned on GadToolsBox will automatically adjust
                       the gadget, text and border coordinates so that
                       they appear to be on the same location.

    Overwrite   When you try to save a file and the file does
                       already exist you will be notified about this
                       with a requester allowing you to overwrite the
                       file or to cancel the operation. When this option
                       is turned on GadToolsBox will not put up the
                       requester.   Instead it will overwrite the file
                       immediatly. USE WITH CARE.

    ASL FileRequester Some people do not seem to like the ReqTools
                       filerequester. I cannot imagine why but turning on
                       this option will tell GadToolsBox to use the ASL
                       filerequester instead of the ReqTools one.

    Font Adaptive   This option affects both the main program and all
                       source generators. When this option is turned on
                       it tells the source generators to generate source
                       that will automatically adapt to the used font.
                       Please refer to @{ " The Generate Source " link source } ←
                          chapter and
                       the @{ " Font Adaptivity " link adaptivity } chapter for ←
                          more info. When
                       this option is turned on GadToolsBox will only
                       allow you to use the topaz 8 font for editing your
                       GUI for reasons explained in the above mentined
                       chapters.

    Close Workbench   When this option is turned on GadToolsBox will
                       attempt to close the Workbench at startup.

    Use PubScreen   When this option is on and you are editing a GUI
                       for the workbench or the default public screen the
                       edit windows will appear on that screen. In other
                       words -: GadToolsBox will not use it's own screen
                                in this case.

    Save   This will save your settings to the GadToolsBox

                              preferences directory.

    Use   This will activate the preferences in the program.

    Cancel   This aborts the operation.

1.21   Project/Preferences/C Source...

This option allows you to control certan aspects of the C source
generator. You will see the following gadgets in the requester:

    Static   Normally all data generated by GadToolsBox will
                        be exported so that it can be accessed from other
                        modules. When this option is on the data generated
                        by GadToolsBox will not be exported.

    Generate OpenFont   When this option is on and you are generating the
                        source for a GUI which uses a disk based font
                        GadToolsBox will also include the code that will
                        open and close the used font.
                        NOTE: This option is not available when the
                                @{ " Font Adaptive " link mainprefs } option is on ←
                                .

    Use System Font   When this option is on the generated source will
                        adapt to the system default font. When this option
                        is off the generated source will use the screen
                        font to adapt to.
                        NOTE: This option is only available when the
                                @{ " Font Adaptive " link mainprefs } option is on ←
                                .

    Include Pragmas   Some C compilers, SAS & Manx, use pragmas to
                        create inline calls to the system libraries. When
                        this option is on the generated source will
                        include the necesarry pragma headers.

    Aztec C   The Manx compiler needs a special type-cast to
                        correctly   initialize the TagItem arrays that
                        GadToolsBox generates. When this option is turned
                        on GadToolsBox will use that type-cast so that the
                        source will also compile using the Manx compiler.

    Generate IDCMP   This option allows you to tell GadToolsBox to also
    Handler   generate an IDCMP handler.   Please refer to
                        @{ " The IDCMP Handler " link handler } chapter for more ←
                           info.

    Generate Templates This option, when switched on, tells GadToolsBox
                        to generate a templates file which contains a set
                        of empty routines that the IDCMP Handler can call.
                        You should use this file as a reference or to fill
                        in with your own code.   This option is only
                        available when the "Generate IDCMP Handler" option
                        is turned on.

```
Save                    This will save your  settings to  the  GadToolsBox
                        preferences directory.

Use                     This will activate the preferences in the program.

Cancel                  This aborts the operation.
```

1.22   Project/Preferences/Asm Source...

This option allows you to control certan aspects of the assembly
source generator. You will see the following gadgets in the requester:

```
Static                  Normally all data  generated  by  GadToolsBox will
                        be exported so that it can be accessed  from other
                        modules. When this option is on the data generated
                        by GadToolsBox will not be exported.

Raw Source              Normally the generated source will need the OS 2.0
                        headers to assemble.   With this option turned on
                        you  are  able  to  assemble  to  generated source
                        without using the headers.

Generate OpenFont       When this option is on and you are  generating the
                        source for a GUI which  uses  a  disk  based  font
                        GadToolsBox will also include the  code  that will
                        open and close the used font.
                        NOTE:   This  option  is not  available  when  the
                                @{ " Font Adaptive " link mainprefs } option is on ←
                                .

Use System Font         When this option is on the  generated  source will
                        adapt to the system default font. When this option
                        is off the  generated  source  will use the screen
                        font to adapt to.
                        NOTE:   This  option  is only available  when  the
                                @{ " Font Adaptive " link mainprefs } option is on ←
                                .

Use amiga.lib           When this option is on all  system  calls  will be
                        externally referenced from the amiga.lib using the
                        _LVOxxxx references.   When this is off the source
                        will contain equates  that  define  the  offsets of
                        the system calls.

Save                    This will save your  settings to  the  GadToolsBox
                        preferences directory.

Use                     This will activate the preferences in the program.

Cancel                  This aborts the operation.
```

1.23   Project/Close Workbench

This menu can also be called "Open Workbench" depending on wether the
the Workbench screen is open or closed.

This will try to close/open the Workbench screen. Closing the Workbench
will only succeed if there are no programs running that have a lock or a
window on the Workbench screen. This also means that when you started
GadToolsBox from the shell you cannot close the Workbench screen because
the shell is a program which has it's window on the Workbench screen.

1.24   Project/About...

This will open a requester with some information in it concerning the
version, copyright and author of GadToolsBox.

1.25   Project/Quit...

You guessed it. This will quit the program.   When there where changes
made that where not saved yet the program will present you with the
following choises:

```
  A)   Save First        First save the GUI, then quit.
  B)   Quit              Go ahead and quit.
  C    Cancel            Do not quit.
```

1.26   Gadgets/Gadgets

During the editing the gadgets are not actually on the editing window.
They are created, refreshed and removed again for the sake of speed and
the selection method.   There may however come a time when you want to
test the actual gadgets. This is where this menu item comes in.

When it is checked the actual gadgets are placed in the window. You can
click on them like you would normally. No editing is posible.

When this item is not checked the program is in the editing mode which
means that you can move, size, delete etc. the gadgets.

1.27   Gadgets/Copy

This will make a copy of all the @{ " selected " link selection } gadgets. All ←
    you have to do
is placing them at the correct location.   To do this you must move the
mouse pointer until the you reach the correct location and press the
left mouse button.

The copied gadgets will have the exact same attributes as their
originals except for the source label.

Please refer to @{ " The Gadget Requesters " link requesters } chapter or more ←
    information.

1.28 Gadgets/Delete...

This will pop up a requester which verify's if you really want to
delete the gadgets.   When you select "Delete" in that requester all
@{ " selected " link selection } gadgets will be erased from memory.

Please be carefull because you cannot get back what you delete!

1.29 Gadgets/Edit...

This will pop up the requester for each @{ " selected " link selection } gadget. ←
     Please refer
to @{ " The Gadget Requesters " link requesters } chapter for more information.

1.30 Gadgets/Join

This will Join @{ " LISTVIEW " link listview } gadgets with @{ " STRING " link ←
    string } gadgets. When you have
@{ " selected " link selection }, for example, three LISTVIEW gadgets and two ←
    STRING gadgets
and you select this item the first two LISTVIEW gadgets will be joined
with the selected string gadgets.

The advantage of joining is that the selection made in the LISTVIEW
will automatically be copied into the STRING gadget. A disadvantage is
that the gadtools.library will force the STRING gadget directly under-
neath the LISTVIEW and it forces the STRING gadget to be of the same
width as the LISTVIEW gadget.

1.31 Gadgets/Split

This is the oposite of @{ " Join " link join }. This will split up all @{ " ←
    selected " link selection } gadgets
again.

1.32 Gadgets/Align/Left

This asks you to click on the gadget that has the left-edge that you want
to align to.   When you click on the gadget all @{ " selected " link selection ←
    } gadgets will
inherrit the left-edge coordinate of the clicked gadget.

1.33 Gadgets/Align/Right

This asks you to click on the gadget that has the right-edge that you want
to align to.   When you click on the gadget all @{ " selected " link selection ←
    } gadgets will
inherrit the right-edge coordinate of the clicked gadget.

NOTE: This might not always work when you align to the right-edge of a
      @{ " PALETTE " link palette } gadget. The gadtools.library sizes this   ←
         gadget so that it
      fits within the area specified which makes it possible that the
      gadget is actually smaller than specified.

1.34 Gadgets/Align/Top

This asks you to click on the gadget that has the top-edge that you want
to align to.   When you click on the gadget all @{ " selected " link selection ←
    } gadgets will
inherrit the top-edge coordinate of the clicked gadget.

1.35 Gadgets/Align/Bottom

This asks you to click on the gadget that has the bottom-edge that you want
to align to.   When you click on the gadget all @{ " selected " link selection ←
    } gadgets will
inherrit the bottom-edge coordinate of the clicked gadget.

NOTE: This might not always work when you align to the bottom-edge of a
      @{ " LISTVIEW " link listview } gadget. The gadtools.library sizes this ←
         gadget so that
      it fits within the area specified which makes it possible that the
      gadget is actually smaller than specified.

1.36 Gadgets/Clone/Width

This will ask you to click on the gadget that has the width that you want
to clone. When you click on the desired gadget all @{ " selected " link ←
    selection } gadgets will
get the same width as the clicked gadget.

1.37   Gadgets/Clone/Height

```
This will ask you to click on the gadget that has the height that  you want
to clone. When you click on the desired gadget  all @{ " selected " link  ←
    selection } gadgets will
get the same height as the clicked gadget.
```

1.38   Gadgets/Clone/Width & Height

```
This will ask you to click on the gadget that has the dimensions  that  you
want to clone. When you click on the desired gadget  all @{ " selected " link  ←
    selection } gadgets
will get the same dimensions as the clicked gadget.
```

1.39   Gadgets/Spacing/Set X...

```
A small requester will popup when you select this option. In this requester
you can enter the horizontal spacing value that you need.   The  value that
you enter here is the space in pixels  that you  want between  the gadgets.
When  the value is entered the space between the @{ " selected " link selection }  ←
    gadgets will be
set.
```

1.40   Gadgets/Spacing/Set Y...

```
A small requester will popup when you select this option. In this requester
you can enter the vertical spacing value that you need.     The  value that
you enter here is the space in pixels  that you  want between  the gadgets.
When  the value is entered the space between the @{ " selected " link selection }  ←
    gadgets will be
set.
```

1.41   Gadgets/Spread/Horizontally

```
When  you  select  this option  you will see a vertical line in  the window
which you can move with the mouse.   This  line  is  used  to set the first
coordinate  of the area in which the @{ " selected " link selection } gadgets   ←
   are  spread. Move
the line to the correct place and press the left mouse butten.

Now a second line appears which is used to set the second coordinate of the
spread area.  Move this line to the correct place  and also press  the left
mouse button.

Now the gadgets are evenly spread out between the two lines.

You  can also  use  this routine  to center a single gadget between the two
lines.
```

1.42   Gadgets/Spread/Vertically

When you select this option you will see a horizontal line in the window
which you can move with the mouse.   This line is used to set the first
coordinate of the area in which the @{ " selected " link selection } gadgets   ←
   are spread. Move
the line to the correct place and press the left mouse butten.

Now a second line appears which is used to set the second coordinate of the
spread area. Move this line to the correct place and also press the left
mouse button.

Now the gadgets are evenly spread out between the two lines.

You can also use this routine to center a single gadget between the two
lines.

1.43   Gadgets/Transform/MX->CYCLE

This routine will convert all @{ " selected " link selection } @{ " MX " link mx ←
   } gadgets into @{ " CYCLE " link cycle } gadgets.

1.44   Gadgets/Transform/CYCLE->MX

This routine will convert all @{ " selected " link selection } @{ " CYCLE " link ←
   cycle } gadgets into @{ " MX " link mx } gadgets.

1.45   Gadgets/Edit TabCycle Order...

This will popup a requester with the following gadgets:

    GadgetText: GadgetLabel: In this   gadget all the @{ " STRING " link ←
       string }   and
                                    @{ " INTEGER " link integer } gadgets are listed ←
                                      .

    Top   This will put the gadget at the top of the
                                    cycle list.

    Bottom   This will put the gadget at the bottom of
                                    the cycle list.

    Up   This will move the gadget one place up in
                                    the cycle list.

    Down   This will move the gadget one place down
                                    in the cycle list.

    OK   Confirm and set the order.

    Cancel   Cancel the operation.

Changing the TabCycle order of the gadgets prevents the random selection of
the next/previous string/integer gadget when the user presses TAB or SHIFT-
TAB in a string or integer gadget.

1.46 Gadgets/Load...

This will put up a filerequester in which you can select the name of a
previously saved gadgets file.

If there are already gadgets in the window GadToolsBox will ask if the
loaded gadgets should be appended to the existing ones or if they should
replace them.

1.47 Gadgets/Save Selected...

This will put up a filerequester in which you can select the name to which
all @{ " selected " link selection } gadgets will be saved.

1.48 Gadgets/Save...

The same as @{ " Save Selected... " link selectedsave } except that now all ←
   gadgets are saved.

1.49 Window/New...

This will open up a blank new edit window in which you can edit another
GUI. This will allow you to create all the windows you need.

1.50 Window/Delete...

A selection window will pop up in which all the windows that currently
are in memory are listed. Just select the window that you want to delete
by double-clicking it or by clicking on it and then on OK.

1.51 Window/Other...

 A selection window will pop up in which all the windows that currently
 are in memory are listed. Just select the window that you want to edit
 by double-clicking it or by clicking on it and then on OK.

1.52   Window/Edit Data...

This opens a window in which you can edit some data concerning the
window. The following gadgets are in the window:

    Project Name   Here you must type in the name of the window.
                            This name must be unique.   This name will be
                            used to built the various source code labels.

    Start ID From   When you plan to share your IDCMP port with
                            other windows in might be a good idea to let the
                            gadget ID's start at a high number so that they
                            do not interfere with the gadgets in the other
                            window. This gadget allows you to set the number
                            at which the gadget ID's begin to count.

    The following gadgets Min X, Min Y, Max X and Max Y are used to set
    the minimum and maximum window dimensions.   They are only available
    for windows which have a size gadget.

    OK   This will confirm your changes.

    Cancel   This will abort the changes.

1.53   Window/Print Info...

This will write some information about the currently active window to the
printer.

1.54   Window/Load...

A FileRequester opens in which you can select the name of a previously
saved window file.   The loaded window will be appended to the already
existing ones.

1.55   Window/Save...

This will open the window selection requester in which you should select
the window you want to save by double-clicking on it or by clicking on it
and then on OK.

Now a FileRequester opens in which you can select the name to save the
window to.

1.56   Window/Edit Flags...

A window will open in which all intuition window flags are listed. You
can select the flags by checking them.

NOTE: Only the SIZEGADGET, CLOSEGADGET, DRAGBAR,   CLOSEWINDOW,
      SIZEBBRIGHT and SIZEBBOTTOM flags are actually used in the program
      during the editing. All the other flags that are turned on will
      only occure in the generated source.

1.57   Window/Edit IDCMP...

A window will open in which all intuition IDCMP flags are listed. You
can select the flags by checking them.

NOTE: None of the IDCMP flags are actually used withing GadToolsBox. The
      flags will only turn up in the generated source.

      The three highlighted flags of in this window must be checked if
      you want the IDCMP handler to generate calls to these routines.

1.58   Window/Edit Tags...

This will open up a window in which you can edit some of the window
attributes. In the window you will find the following gadgets:

    InnerWidth   When this gadget is checked the generated source
                          will open the window using the WA_InnerWidth tag
                          instead of the WA_Width tag.   This tag is useful
                          to always open a window with the same inner-
                          dimensions no matter what the screen font is.

    InnerHeight   When this gadget is checked the generated source
                          will open the window using the WA_InnerHeight tag
                          instead of the WA_Height tag.   This tag is useful
                          to always open a window with the same inner-
                          dimensions no matter what the screen font is.

    MouseQueue   This enables you to set the maximum # of MOUSEMOVE
                          messages that will be pending at your message port
                          at one time.

    RptQueue   The same as MouseQueue only this time the maximum
                          # of RAWKEY/VANILLAKEY messages can be set.

    AutoAdjust   When   this   gadget   is checked intuition will
                          automatically   adjust   the   window position and
                          dimensions to force it to fit on the screen.

    FallBack   When your window is supposed to open on a specific
                          screen and that screen is not available it will
                          automatically open on the default public screen
                          when this gadget is checked.

```
Prefs Zoom              These two  options  are  only  available  when the
Pos.  Zoom              window does not have a sizing gadget.  It will add
                        a zoom gadget to let the user iconify  the window.
                        The  iconified dimensions  are  the  same  as  the
                        system preferences windows.    When Prefs Zoom is
                        selected the iconified  window  will automatically
                        shift to the top-left corner of the screen.   With
                        Pos. Zoom the window position will not be affected

WindowTitle             Type the window it's title in here.

ScreenTitle             You can type an alternate screen  title  in  here.
                        This will be the  title  of  the  screen  when  the
                        window is active.

OK                      Confirm the attributes.

Cancel                  Cancel the operation.
```

1.59   Window/Edit Grid...

```
This will open a window in which you can edit the grid size.    The grid,
which is invisible, is used to let the gadgets,  boxes  and  texts to  be
moved and sized with a specific amount of pixels at the time.    The grid
window has the following gadgets in it:

    Grid X size             In here you can type the horizontal grid size. You
                            are not allowed to go below 2 and over 100 here.

    Grid Y size             In here you can type the vertical grid size.   You
                            are not allowed to go below 2 nd over 100 here.

    Grid on                 When you check this gadget the grid is active.

    OK                      Confirm the grid.

    Cancel                  Cancel the operation.
```

1.60   Window/Edit Offsets...

```
This will open a window in which you can  edit  the  minimal window  border
offsets. The border offsets are the closest you can put  your  gadgets  and
boxes to the window borders. In the window you see the following gadgets:

    Horiz. Offset           Here you can type the  horizontal  border  offset in
                            pixels. This is the offset from  the left  and right
                            window borders.

    Vert. Offset            Here you can type  the  vertical  border  offset  in
                            pixels. This is the offset from  the top  and bottom
                            window borders.
```

```
    Offsets On            When this gadget is checked the offsets are active.

    OK                    Confirm the offsets.

    Cancel                Cancel the operation.
```

1.61 Window/Texts/Add...

This will open up the IntuiText editor in which you can edit a new text.
You will see the following gadgets in the IntuiText editor:

```
    Text                  Type the text that you want in here.

    JAM1                  These four gadgets determine the draw  mode  that is
    JAM2                  used to render the text.
    COMPLEMENT
    INVERSVID

    FrontPen              With this gadget you can set the color which is used
                          to render the text.

    BackPen               With this gadget you can set the color which is used
                          to render the text background.

    OK                    Confirm the text.

    Cancel                Cancel the operation.
```

After the text is editied you will see that the text is following the
movements you make with your mouse. Now position the text where you want it
and click on the left mouse button to confirm it's place.

1.62 Window/Texts/Edit...

When there are more than one texts available in the window a text selection
window will pop up from which you must select a text. The following gadgets
are in the text selector:

```
    Texts                 This is the list of available texts.   To select one
                          you either double-click it or you  click on  it once
                          and then on OK.

    OK                    Confirm the choosen text.

    Cancel                Cancel the operation.
```

When you have selected the text the @{ " IntuiText editor " link addtext } pops ←
     up in which
you can edit the text.

1.63   Window/Texts/Delete...

```
If there is more than one text available in the window  the @{ " text selector "  ←
    link edittext }
opens in which you must select the text. After you selected the  text  that
needs to be deleted it is gone.

When only one text is available it is deleted right away.
```

1.64   Window/Texts/Move...

```
If there is more than one text available in the window  the @{ " text selector "  ←
    link edittext }
opens in which you must select the text.

Now you see the text following the movements you make with your mouse.  Now
move the text to it's new position and click  on the  left mouse  button to
confirm it's new place.
```

1.65   Window/Texts/Load...

```
This will open up a filerequester in which you can  select  the  name of  a
previously saved text file.

After the loading is done you will see the texts that where in the  file on
your window.
```

1.66   Window/Texts/Save...

```
This will open up a filerequester in which you must  select  a  name  which
will be used to save the texts in the window.
```

1.67   Window/BevelBoxes/Add

```
You will be asked to click at the top-left position where you want the box
to be. After that you must drag the box until it is the  correct  size and
release the left mouse button again.
```

1.68   Window/BevelBoxes/Move

```
You will be asked to click in the box that you want to move.    Just click
in the box with the left mouse button.

Now move the box to it's new position and release  the left  mouse  button
to confirm it's place.
```

1.69   Window/BevelBoxes/Size

You will be asked to click in the box that you want to re-size. Just
click in the box with the left mouse button.

Now move the mouse to re-size the box and release the left mouse button
to confirm it's new dimensions.

1.70   Window/BevelBoxes/Delete

You will be asked to click in the box that you want to delete. Just click
in the box with the left mouse button and it is gone.

1.71   Window/BevelBoxes/Flip Recessed

By default all created boxes appear raised. To make it appear recessed you
select this item and click in the box.

When the box was already recessed it will become raised again.

1.72   Window/BevelBoxes/Flip DropBox

This will convert a normal or recessed box into an Icon DropBox. The usage
of an Icon DropBox is described in the User Interface Style Guide.

If the box was already an Icon DropBox it will become a normal box again.

1.73   Window/BevelBoxes/Load...

This will open up a filerequester in which you can select the name of a
previously saved BevelBox file.

After the loading is done you will see the boxes that where in the file on
your window.

1.74   Window/BevelBoxes/Save...

This will open up a filerequester in which you must select a name which
will be used to save the boxes in the window.

1.75   Screen/Palette...

This will open up the palette editor for you to change the screen colors.
In the palette editor you will see the following gadgets:

   Red, Green, Blue   With these three sliders you can adjust the red,
                    green and blue intensities of the selected color.

   Load...   This will open up a filerequester in which you can
                    an IFF-ILBM file from which you want to load it's
                    colors.

   Save...   This will open up a filerequester in which you can
                    select a name. This name will be used to save the
                    current palette under as an IFF-ILBM file.

   Reset   Delete all changes made sofar and reset the palette
                    to what it was when you first entered the palette
                    editor.

   Palette   This is the color selection gadget. In here you can
                    select the color that you wish to edit.

   OK   Confirm the palette.

   Cancel   Cancel the operation.

 NOTE: Under Kickstart 3.0 the palette editor has a problem.   Although it
      shows more than 32 colors on a screen with more than 5 planes it will
      only edit the first 32.

1.76   Screen/Select Font...

This will open up the Font Selector in which you can select the new font.

The windows in your GUI will automatically be adapted to the new font
so that the windows still look like they should.

1.77   Screen/Edit DriPens...

This will open the DriPen editor. The DriPens are an array of pen numbers
which are used to create the OS 2.0 new look.

The following gadgets are in the DriPen Editor:

   Pens   In this cycle gadget you can select the pen to edit.
                    Currently the following pens are supported:

                    DETAILPEN
                        1.3 compatible DetailPen.

                    BLOCKPEN

```
                            1.3 compatible BlockPen.

                   TEXTPEN
                        Used for rendering gadget texts and  the  window
                        titles.

                   SHINEPEN
                        Used to render  the  light-side  of  the  gadget
                        and window borders.

                   SHADOWPEN
                        Used to render the dark-side of  the  gadget and
                        window borders.

                   FILLPEN
                        Used to render  the  background  of  a  selected
                        gadget and the window borders when it is active.

                   FILLTEXTPEN
                        Used to render  the  gadget  texts  of  selected
                        gadgets and window titles of active windows.

                   BACKGROUNDPEN
                        Used to render  the  background  of  un-selected
                        gadgets.

                   HIGHLIGHTTEXTPEN
                        Used to render the  texts  of  gadgets  with the
                        Highlight switch on.

                   NOTE: The following three pens  are  only  available
                        under 3.0.

                   BARDETAILPEN
                        Used to render the text of the NewLook menus.

                   BARBLOCKPEN
                        Used to render  the  background  of the  NewLook
                        menus.

                   BARTRIMPEN
                        Used to render the line below the menu bar.

   Palette               With this gadget you can change the actual DriPen.

   OK                    Confirm the DriPens.

   Cancel                Cancel the operation.
```

1.78 Screen/Edit Tags...

```
 In the Screen Tags editor you will find the following gadgets.

   Type                  This gadget will allow you to change the screen type
                         you are working for.     The  following  types  are
```

                              supported:

                              CUSTOMSCREEN
                                   The generated source will contain the screen
                                   tags and data. The windows will use this screen
                                   to open on.

                              WBENCHSCREEN
                                   The generated source will not contain the screen
                                   tags and data. The windows will use the WBench
                                   screen to open on.

                              PUBLICSCREEN
                                   The generated source will not contain the screen
                                   tags and data. The windows will use the default
                                   public screen to open on.

     Title   In this gadget you can type the title of the screen.
                              This is the title which will be displayed when there
                              are no windows active which have their own screen
                              title.

     OK   Confirm the tags.

     Cancel   Cancel the operation.

1.79 Screen/Change Type...

This will open the ReqTools screen mode requester.   In the screen mode
requester you can change all important stuff concerning the screen.

When you are running using the workbench or default public screen this
option is still available. Visualy it will not have any effect but when you
switch to not using the workbench or default public screen the changes will
be clear.

THE VGA AND A2024 MODES MAY ONLY BE USED WITH VGA/MULTISYNC OR A2024
MONITORS! USAGE OF THESE DISPLAY MODES ON A STANDARD RGB MONITOR MAY RESULT
IN SERIOUS DAMAGE TO THE MONITOR.

1.80 Menus/Test

This will setup the created menus for you to take a look at them. When the
message "TESTING MENUS! ESC TO QUIT..." appears in the screen title you
can browse to the menus like you normally do.

Pressing on ESC will continue the program.

1.81 Menus/Load...

This will open up a filerequester in which you can select the name of a
previously saved menu file.

After the loading is done the menus are attached to your window.

1.82   Menus/Save...

This will open up a filerequester in which you must select a name which
will be used to save the menus of the window.

1.83   Gadget Requesters

Please select any of the gadget kinds you see listed below.

```
    @{ " GETFILE  " link getfile }
    @{ " BUTTON   " link button }
    @{ " CHECKBOX " link checkbox }
    @{ " INTEGER  " link integer }
    @{ " LISTVIEW " link listview }
    @{ " MX       " link mx }
    @{ " NUMBER   " link number }
    @{ " CYCLE    " link cycle }
    @{ " PALETTE  " link palette }
    @{ " SCROLLER " link scroller }
    @{ " SLIDER   " link slider }
    @{ " STRING   " link string }
    @{ " TEXT     " link text }
```


1.84   GETFILE

This is currently the only custom gadget that is   supported   by
GadToolsBox.

It is ment to be used in conjunction with a @{ " STRING " link string } gadget to ←
    enable the
user to select a file/dir/volume name. When this gadget is selected the
program should open a filerequester in which the file/dir/volume can be
selected. As a result the selected name should be copied into the STRING
gadget.

The attribute requester of this gadget kind allows you to edit the
following attributes:

    Label   In here the source-code name of the gadget should
                      be entered. The source code name is important to
                      recognize the gadget later on in the generated
                      source.

    Disabled   When this option is on the created gadget will be

                                        created disabled. NOTE: This gadget kind does not
                                        support   the   enabling/disabling by a   call to
                                        GT_SetGadgetAttrs().

    OK   Confirm the attributes.

    Cancel   Cancel the operation.

1.85   BUTTON

This gadget kind, also known as action gadget, is   ment   to   trigger   a
program action. The action it is supposed to trigger is usually described
in the gadget it's text.

The attribute requester of this   gadget   kind   allows   you   to   edit   the
following attributes:

    Text   In here the text of the gadget can be typed.   The
                                        text usually describes the gadget function   in one
                                        to three words. When the gadget should open   up   a
                                        requester with more options the   text   should   end
                                        with an ellipsis. E.g. "...".

    Label   In here the source-code name of the gadget   should
                                        be entered. The source code name   is   important to
                                        recognize the   gadget   later on   in the   generated
                                        source.

    Place   This gadget enables you   to   determine   where   the
                                        text should be rendered.

    Underscore   When this gadget is checked the character from the
                                        text that is preceeded by a '_' sign will be
                                        underlined. The underlined character usually shows
                                        the key that also activates the gadget.

    Disabled   When this option is on the created gadget   will be
                                        created disabled.

    OK   Confirm the attributes.

    Cancel   Cancel the operation.

1.86   CHECKBOX

This gadget kind is switch which can be toggled on or of.   This kind is
normally used to switch on or off program options.

The attribute requester of this   gadget   kind   allows   you   to   edit   the
following attributes:

    Text   In here the text of the gadget can be   typed.   The

```
                              text usually describes the gadget function  in one
                              to three words.

     Label                    In here the source-code name of the gadget  should
                              be entered. The source code name  is  important to
                              recognize the  gadget  later on  in the  generated
                              source.

     Place                    This gadget enables you  to  determine  where  the
                              text should be rendered.

     Underscore               When this gadget is checked the character from the
                              text that is preceeded  by  a  '_'  sign  will  be
                              underlined. The underlined character usually shows
                              the key that also activates the gadget.

     Disabled                 When this option is on the created gadget  will be
                              created disabled.

     Highlabel                When this gadget  is  checked  the  text  will  be
                              rendered using the HIGHLIGHTTEXTPEN color from the
                              screen @{ " DriPens " link dripens }.

     Checked                  By default the created gadget is  off.   When  you
                              turn this option on  the  gadget  will  be created
                              "checked".

     OK                       Confirm the attributes.

     Cancel                   Cancel the operation.
```

1.87 INTEGER

This gadget kind is normally used to obtain numeric values from the user.

The attribute requester of this gadget kind allows you to edit the
following attributes:

```
     Text                     In here the text of the gadget can be  typed.  The
                              text usually describes the gadget function  in one
                              to three words.

     Label                    In here the source-code name of the gadget  should
                              be entered. The source code name  is  important to
                              recognize the  gadget  later on  in the  generated
                              source.

     Place                    This gadget enables you  to  determine  where  the
                              text should be rendered.

     Underscore               When this gadget is checked the character from the
                              text that is preceeded  by  a  '_'  sign  will  be
                              underlined. The underlined character usually shows
                              the key that also activates the gadget.
```

Disabled   When this option is on the created gadget will be
                     created disabled.

Highlabel   When this gadget is checked the text will be
                     rendered using the HIGHLIGHTTEXTPEN color from the
                     screen @{ " DriPens " link dripens }.

ExitHelp   When this gadget is checked and the user presses
                     the HELP key while the created gadget is active
                     your program will receive a special message at the
                     the window port telling you about the quest for
                     help. You can identify this special message when
                     the Code field of the IntuiMessage is 0x5F.
                     Normally you display some information as to the
                     usage of the gadget when you receive such a
                     message.

TabCycle   Normally all @{ " STRING " link string } and INTEGER ←
   gadgets are in
                     the @{ " TABCYCLE " link tabcycle } list. This list makes ←
                        sure that the
                     next/previous   STRING   or INTEGER   gadget   is
                     activated when TAB/SHIFT-TAB is pressed in the
                     gadget. When you turn this gadget off the created
                     gadget will not be included in this list.

Placement   With this gadget you can set the place where the
                     numbers should appear in the gadget.

MaxChars   In this gadget you can set the maximum number of
                     digits that can be entered in this gadget.

Number   Normally the gadget is created with 0 as it's
                     initial value. Here you can enter the value that
                     you want it to default to.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.88 LISTVIEW

This gadget kind is normally used to display a list of textual names.
This list can either be a list of names from which one can be selected or
a list of names for information purposes.

The attribute requester of this gadget kind allows you to edit the
following attributes:

Text   In here the text of the gadget can be typed. The
                     text usually describes the gadget function in one
                     to three words.

Label   In here the source-code name of the gadget should
                     be entered. The source code name is important to

recognize the gadget later on in the generated
source.

Place   This gadget enables you to determine where the
                          text should be rendered.

Underscore   When this gadget is checked the character from the
                          text that is preceeded by a '_' sign will be
                          underlined. The underlined character usually shows
                          the key that also activates the gadget.

Read Only   When you switch this on the created gadget will be
                          "read only" which means that the names in the list
                          cannot be selected.

Highlabel   When this gadget is checked the text will be
                          rendered using the HIGHLIGHTTEXTPEN color from the
                          screen @{ " DriPens " link dripens }.

ShowSelected   With this option turned on the name you select
                          will be shown in a little box below the actual
                          gadget. NOTE: Under Kickstart 3.0 this has been
                          changed! Under 3.0 the selected entry it's
                          background color will change. You can also tell it
                          to copy the selected name into a @{ " joined " link join } ←
                              @{ " STRING " link string }
                          gadget.

Spacing   In this gadget you can enter the number of pixels
                          that is used as spacing between the list names.

Scr. Width   In this gadget you can enter the width of the
                          @{ " SCROLLER " link scroller } gadget in pixels.

List   This entry list allows you to enter and edit names
Add   that occure in the created gadget.   With Add you
Del   add a new name to the list.   With Del you can
Up   delete a name from the list. With Up and Down you
Down   change a name it's position in the list.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.89 MX

This gadget kind is used to let the user select one out of several
possibilities.

The attribute requester of this gadget kind allows you to edit the
following attributes:

Label   In here the source-code name of the gadget should
                          be entered. The source code name is important to
                          recognize the gadget later on in the generated

source.

Place   This gadget enables you to determine where the
                    text should be rendered.

Underscore   When this gadget is checked the character from the
                    text that is preceeded by a '_' sign will be
                    underlined. The underlined character usually shows
                    the key that also activates the gadget.

Spacing   In this gadget you can enter the number of pixels
                    that is used as spacing between the options.

Active   This gadget enables you to set the option that is
                    active when the gadget is created.

List   This entry list allows you to enter and edit names
Add   that occure in the created gadget.   With Add you
Del   add a new name to the list.   With Del you can
Up   delete a name from the list. With Up and Down you
Down   change a name it's position in the list.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.90 NUMBER

This is a read only gadget which is ment to be used to display a
numerical value. It has a informative function.

The attribute requester of this gadget kind allows you to edit the
following attributes:

Text   In here the text of the gadget can be typed. The
                    text usually describes the gadget function in one
                    to three words.

Label   In here the source-code name of the gadget should
                    be entered. The source code name is important to
                    recognize the gadget later on in the generated
                    source.

Place   This gadget enables you to determine where the
                    text should be rendered.

Border   When checked the created gadget will have a
                    recessed border drawn arround it.

Default   By default the gadget displays 0. In here you can
                    change the value that the created gadget will
                    default to.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.91 CYCLE

This gadget kind is used to let the user select one out of several
possibilities.

The attribute requester of this gadget kind allows you to edit the
following attributes:

    Text   In here the text of the gadget can be typed.   The
                        text usually describes the gadget function in one
                        to three words.

    Label   In here the source-code name of the gadget should
                        be entered. The source code name is important to
                        recognize the gadget later on in the generated
                        source.

    Place   This gadget enables you to determine where the
                        text should be rendered.

    Underscore   When this gadget is checked the character from the
                        text that is preceeded by a '_' sign will be
                        underlined. The underlined character usually shows
                        the key that also activates the gadget.

    Disabled   When this option is on the created gadget will be
                        created disabled.

    Highlabel   When this gadget is checked the text will be
                        rendered using the HIGHLIGHTTEXTPEN color from the
                        screen @{ " DriPens " link dripens }.

    Active   This gadget enables you to set the option that is
                        active when the gadget is created.

    List   This entry list allows you to enter and edit names
    Add   that occure in the created gadget.   With Add you
    Del   add a new name to the list.   With Del you   can
    Up   delete a name from the list. With Up and Down you
    Down   change a name it's position in the list.

    OK   Confirm the attributes.

    Cancel   Cancel the operation.

1.92 PALETTE

This gadget kind normally is used to let the user select a color from a
set palette.

The attribute requester of this gadget kind allows you to edit the following attributes:

Text   In here the text of the gadget can be typed. The text usually describes the gadget function in one to three words.

Label   In here the source-code name of the gadget should be entered. The source code name is important to recognize the gadget later on in the generated source.

Place   This gadget enables you to determine where the text should be rendered.

Underscore   When this gadget is checked the character from the text that is preceeded by a '_' sign will be underlined. The underlined character usually shows the key that also activates the gadget.

Disabled   When this option is on the created gadget will be created disabled.

Highlabel   When this gadget is checked the text will be rendered using the HIGHLIGHTTEXTPEN color from the screen @{ " DriPens " link dripens }.

Width   When this gadget is checked the created gadget will have an indicator to the left. The indicator is used to display the currently selected color. The size of the indicator can be changed in the @{ " INTEGER " link integer } gadget.

Height   When this gadget is checked the created gadget will have an indicator on top. The indicator is used to display the currently selected color. The size of the indicator can be changed in the INTEGER gadget.

Depth   The depth allows you to control the amount of colors that are displayed in the created gadget.

                            1 = 2 colors
                            2 = 4 colors
                            ......
                            8 = 256 colors

                    NOTE: Currently only a value between 1 and 8 is
                          allowed.

Color   In this gadget you can enter the color that is active in the created gadget.

Offset   There may be times that you only want to have a portion of the available colors in the palette gadget. This gadget and the Depth gadget can make

this happen. If you want to have the 2nd and 3rd
color on a 4 color screen you do the following:

```
Set Depth to 1
Set Offset to 1
```

This way you'll get the gadget you want.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.93 SCROLLER

This gadget kind is used to adjust the positions of a view.   For example
scrolling the text up and down in a text editor.

The attribute requester of this gadget kind allows you to edit the
following attributes:

Text   In here the text of the gadget can be typed. The
                        text usually describes the gadget function in one
                        to three words.

Label   In here the source-code name of the gadget should
                        be entered. The source code name is important to
                        recognize the gadget later on in the generated
                        source.

Place   This gadget enables you to determine where the
                        text should be rendered.

Underscore   When this gadget is checked the character from the
                        text that is preceeded by a '_' sign will be
                        underlined. The underlined character usually shows
                        the key that also activates the gadget.

Disabled   When this option is on the created gadget will be
                        created disabled.

Highlabel   When this gadget is checked the text will be
                        rendered using the HIGHLIGHTTEXTPEN color from the
                        screen @{ " DriPens " link dripens }.

Immediate   When this option is on your window will receive a
                        message at the moment the user clicks on the
                        gadget.

RelVerify   When checked your window will receive a message
                        when the user releases the left mouse button while
                        the pointer is still over the gadget.

Freedom   With this gadget you can set the direction in
                        which the SCROLLER is adjustable.

Top   Enter the top position visible in the area the
                        SCROLLER represents in here.

Total   Enter the total number of positions that the
                        SCROLLER represents in here.

Visible   Enter the number of visible positions that the
                        SCROLLER represents in here.

Arrows   When checked the created SCROLLER will have arrows
                        to accompany it. In the gadget on the right you
                        can adjust the size of the arrows.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.94 SLIDER

This gadget kind is used to select a value within a fixed range. For
instance the red, green and blue intensities of a palette editor.

The attribute requester of this gadget kind allows you to edit the
following attributes:

Text   In here the text of the gadget can be typed. The
                        text usually describes the gadget function in one
                        to three words.

Label   In here the source-code name of the gadget should
                        be entered. The source code name is important to
                        recognize the gadget later on in the generated
                        source.

Place   This gadget enables you to determine where the
                        text should be rendered.

Underscore   When this gadget is checked the character from the
                        text that is preceeded by a '_' sign will be
                        underlined. The underlined character usually shows
                        the key that also activates the gadget.

Disabled   When this option is on the created gadget will be
                        created disabled.

Highlabel   When this gadget is checked the text will be
                        rendered using the HIGHLIGHTTEXTPEN color from the
                        screen @{ " DriPens " link dripens }.

Immediate   When this option is on your window will receive a
                        message at the moment the user clicks on the
                        gadget.

RelVerify   When checked your window will receive a message
                     when the user releases the left mouse button while
                     the pointer is still over the gadget.

Freedom   With this gadget you can set the direction in
                     which the SLIDER is adjustable.

Min   Enter the minimum level of the SLIDER in here.

Max   Enter the maximum level of the SLIDER in here.

Level   By default the position of the SLIDER is 0.   In
                     here you can enter the level that the SLIDER will
                     have when it is created.

Length   In this gadget you must enter the maximum amount
                     of characters that are displayed by the level
                     indicator.

Format   In here you can enter a printf() style formatting
                     string for the level indicator. When you keep this
                     empty there will not be a level indicator.

Position   With this gadget you determine the position at
                     which the level indicator is rendered.

OK   Confirm the attributes.

Cancel   Cancel the operation.

1.95 STRING

This gadget kind is normally used to obtain character strings from the
user.

The attribute requester of this gadget kind allows you to edit the
following attributes:

Text   In here the text of the gadget can be typed. The
                     text usually describes the gadget function in one
                     to three words.

Label   In here the source-code name of the gadget should
                     be entered. The source code name is important to
                     recognize the gadget later on in the generated
                     source.

Place   This gadget enables you to determine where the
                     text should be rendered.

Underscore   When this gadget is checked the character from the
                     text that is preceeded by a '_' sign will be
                     underlined. The underlined character usually shows
                     the key that also activates the gadget.

```
     Disabled            When this option is on the created gadget  will be
                         created disabled.

     Highlabel           When this gadget is  checked  the  text  will  be
                         rendered using the HIGHLIGHTTEXTPEN color from the
                         screen @{ " DriPens " link dripens }.

     ExitHelp            When this gadget is checked and  the  user presses
                         the HELP key while  the  created  gadget is  active
                         your program will receive a special message at the
                         the window port telling  you  about the  quest for
                         help. You can identify this special  message  when
                         the  Code  field  of  the IntuiMessage is 0x5F.
                         Normally you display  some  information as  to the
                         usage of  the  gadget  when  you  receive  such  a
                         message.

     TabCycle            Normally all STRING and @{ " INTEGER " link integer }  ←
        gadgets   are  in
                         the @{ " TABCYCLE " link tabcycle } list. This list makes  ←
                            sure that the
                         next/previous   STRING   or   INTEGER   gadget   is
                         activated when  TAB/SHIFT-TAB  is  pressed  in the
                         gadget. When you turn this gadget off the  created
                         gadget will not be included in this list.

     Placement           With this gadget you can set the  place  where the
                         characters should appear in the gadget.

     MaxChars            In this gadget you can set  the maximum  number of
                         characters that can be entered in this gadget.

     String              Normally the gadget  is  created  without a string
                         in it. Here you can enter  a string  that will  be
                         the contents of the created gadget.

     OK                  Confirm the attributes.

     Cancel              Cancel the operation.
```

1.96 TEXT

This is a read only gadget which is ment to be used to display a
character string. It has a informative function.

The attribute requester of this gadget kind allows you to edit the
following attributes:

```
     Text                In here the text of the gadget can be  typed.  The
                         text usually describes the gadget function  in one
                         to three words.

     Label               In here the source-code name of the gadget  should
                         be entered. The source code name  is  important to
                         recognize the  gadget  later on  in the  generated
```

                           source.

    Place   This gadget enables you to determine where the
                        text should be rendered.

    Border   When checked the created gadget will have a
                        recessed border drawn arround it.

    CopyText   When checked GadTools will copy the text to an
                        internal buffer so that the program can use it
                        again.

    Default   By default the gadget displays nothing.   In here
                        you can type in a text that will be displayed when
                        the gadget is created.

    OK   Confirm the attributes.

    Cancel   Cancel the operation.

1.97 The Menu Editor

The menu editor of GadToolsBox allows you to edit a complete menu-strip
for your windows. The working of the editor is based on a simple
principle. The right-most list with an active item is always the active
input list.

This means that when there is a subitem active the mutual exclude, flags
etc. you change are done on that subitem.

When there is no subitem but an item active the changes are made on
the item.

The following gadgets are in the Menu Editor.

    Menus, Items, SubItems   These three input lists are   used to
                                    add and delete new entries in the
                                    menu. "Add" will add a new entry.
                                    "Del" removes the currently selected
                                    entry. "Up" and "Down" enables you to
                                    move the entries in the list.

    DISABLED, CHECKIT, CHECKED,   These are the menu flags which you can
    MENUTOGGLE   toggle on or off.

    COMMKEY   In here you can type the shortcut
                                    character of the (sub)item.

    BarLabel   When you click on this gadget an
                                    NM_BARLABEL item is inserted in the
                                    (sub)item list.

    MutEx   This will enable you to set the mutual
                                    exclude of a (sub)item. You will be
                                    asked to drag-select the (sub)items

that need to be excluded. You drag-
select by clicking the left mouse
button on the (sub)item whilst holding
down the right mouse button.

Label   In here you can enter the source code
                               name of the (sub)item.

OK   Confirm the menu.

Cancel   Cancel the operation.

1.98   The Keyboard & Mouse

There are few keys which perform a more or less important function
during editing. The following is a short description of the keys:

   When you are draging the rubberband box to add, move, copy or size one
   or more gadgets or when you add, move, size a bevelbox:

   Key: Action:
   ----
   x   This key, while pressed, will block
                               the horizontal movements of the rubber
                               band box.   The movements will be
                               released when you release the key.

   y   The same as the 'x' key only now the
                               vertical movements are blocked.

   ESC or Right Mouse-Button   This will cancel the operation you are
                               doing.

   When you are in the edit window and there is no requester active:

   Key: Action:

   F10   This flips the window sizing gadget on
                               or off. This is simply a shortcut to
                               the WFLG_SIZEGADGET flag.

   F7   This will activate the old '.G' file
                               type reader. It is the same as opening
                               a .GUI file only now a '.G' file is
                               expected.   NOTE: Now GTB   will
                               determine automatically what file to
                               load when selecting "Open..." from the
                               menu.

   a   Select all gadgets in the edit window.

   Shift + a   De-select all gadgets in the edit
                               window.

```
    Shift + Left Mouse-Button        This put's you in drag-selection mode.
                                     Please look  at the @{ " Gadget Selection " ←
                                        link selection }
                                     chapter for more info.


    When you are prompted via the screen and window title to do something:


    Key:                             Action:
    ------------------------------------------------------------------------
    ESC or Right Mouse-Button        Cancel the operation.


    Double-Clicking a gadget will open it's attribute editor.
```

1.99   The GetFile BOOPSI Image

```
When you are using the @{ " GETFILE " link getfile } gadget in your GUI  you * ←
   must* link with
the supplied object code "boopsi.o".   This  contains  the dispatcher and
initialization code that will setup and render the image.

The source code is also included so you can see how it has been done. The
code is reentrant so it can be used in resident programs and libraries.

Please  note  that  this  code  is  *NOT*  public  domain.   It is freely
distributable and remains (C) Copyright 1992-1993 Jaba Development.

You are allowed to distribute it with  your  program no  matter  if it is
a public domain, a freeware, a shareware or a commercial product.
```

1.100   The Generated Source

```
**************************************************************************
* Currently there are three builtin source generators. C, Assembler and *
* Oberon. The next release  will not  have  builtin  source  generators *
* anymore. By then all the  source  generators  should  be  external to *
* create a more flexible environment.                                   *
*                                                                       *
* If you study the "gadtoolsbox.library" docs  and the  source  to the *
* sample source generator you can also build your own.                  *
**************************************************************************


The source output of GadToolsBox is divided into two  separate files. The
first file is a header (.h|.i)  containing  EXTERNS or XREF'S to the data
in the  generated main source and  CONSTANTS  for  the gadget ID's  and
the positions of the gadgets in the  Gadget  Array.

The  second  file  (.c|.s) (from  here  on  referred  to as "main file")
contains all data and routines to set-up things.

From here on when I refer to <Project_Name>  I mean the name  entered  in
the @{ " Edit Data " link datawindow }  requester  from  the  Window  menu  and  ←
   when I refer to
```

```
<Gadget_Label> I mean  the   name  you  have  typed  in the "Label"  gadget
of one  of the  @{ " gadget kind requesters " link requesters }.   When I refer  ←
   to   <Menu_Label>
I mean the label you have typed in the Label  gadget of the @{ " Menu Editor "  ←
   link menueditor }
```

In the main file the following routines are generated:

```
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» long SetupScreen( void );                                                 »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
 This routine will open or lock the screen and get it's visual info.
 When requested this routine will also  open the font.  Also it will
 setup the boopsi image for the @{ " GETFILE " link getfile } gadget when  such ←
    a gadget
 is used.  This  routine  can  return  one of the following errors:

     0 = No error.
     1 = Could not open or lock the screen.
     2 = Could not get the screen it's visual info.
     3 = Could not set-up the GETFILE boopsi class.
     4 = Could not get a GETFILE boopsi object.
     5 = Could not open the font.


»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» void CloseDownScreen( void );                                             »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
 This routine will free the resources taken by "SetupScreen()".  You
 must still call this routine when  "SetupScreen()"  failed to close
 and free the resources that didn't fail to open!!!


»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» long Open<Project_Name>Window( void );                                    »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
 There are as much of these routines as there are Project Windows in
 the file you have generated. These routines will set-up the gadgets
 and menus and open the window. These routines can return one of the
 following errors:

     0 = No error.
     1 = Could not create a gadget context.
     2 = Error during the gadget creation.
     3 = Could not create the menus.
     4 = Could not open the window.
     5 = Could not open the system font.


»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» void Close<Project_Name>Window( void );                                   »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
 There are as much of these routines as there are Project Windows in
 the file you have generated. These routines will free all resources
 that the "Open<Project_Name>Window()" routine has taken.  You  must
 still call this routine if "Open<Project_Name>Window()" has failed!


»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» void <Project_Name>Render( void );                                        »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
```

Each Project Window which has texts and/or BevelBoxes attached to
it will have a routine which renders these texts and/or BevelBoxes
in the window. This routine must also be called by the user program
upon receiving a IDCMP_REFRESHWINDOW message at the window port.
When such a message is received this routine must be called
between the GT_BeginRefresh() and GT_EndRefresh() calls. Here
is a small example of what I mean:

```
    while( 1 ) {
        WaitPort( <Project_Name>Wnd->UserPort );
        while( imsg = GT_GetIMsg( <Project_Name>Wnd->UserPort )) {

            Class = imsg->Class;
            GT_ReplyIMsg( imsg );

            switch ( Class ) {

                case    IDCMP_REFRESHWINDOW:
                    GT_BeginRefresh( Project_Name>Wnd );

                    /* re-draw texts and boxes */
                    <Project_Name>Render();

                    GT_EndRefresh( <Project_Name>Wnd, TRUE );
                    break;

                .....
            }
        }
    }
```

Note that the above is automatically done by the generated IDCMP
event-handler.

The main file also has the following globals defined which are shared by
all project windows:

```
struct Screen *Scr;          A pointer to the opened/locked screen
APTR          VisualInfo;  A pointer to the visual info
```

The following globals are only generated when the file contains gadgets
or menus and when the OpenFont routine is generated and when the window
must have a zoom gadget:

```
struct Gadget   *<Project_Name>GList;  The created gadgets
struct Menu     *<Project_Name>Menus;  The created menus
WORD             <Project_Name>Zoom[4]; The window alternate positions
struct TextFont *Font;                 The opened font
```

Also generated are pointers for each project window that is in memory.
Also the window title is generated as a global pointer.   These pointers
are generated as follows:

```
struct Window   *<Project_Name>Wnd;   A pointer to the window
UBYTE           *<Project_Name>Wdt;   A pointer to the window title
```

When you have generated source for the Workbench or Default Public screen

a global is generated which is a pointer to the name of the public screen
to open on. It is generated as follows:

    When generated for the Workbench screen:

    UBYTE   *PubScreenName = "Workbench";

    When generated for the Default Public screen:

    UBYTE   *PubScreenName = NULL;

This will allow you to change the name of the public screen to open on
with out having to hack in the generated source.

Four words are generated for each window. These words contain the window
it's left and top-edge and the window it's width and height.   These
globals are build as follows:

    UWORD   <Project_Name>Left;   Window left-edge
    UWORD   <Project_Name>Top;   Window top-edge
    UWORD   <Project_Name>Width;   Window width
    UWORD   <Project_Name>Height;   Window height

These words   should be changed by the application if the user has
re-sized or re-positioned the window. This way when the window is closed
and then re-opened again it will be opened the size and position it had
the last time it was open. Note: When the user has changed the window you
must substract the window top-border size from the window height before
setting in the "<Project_Name>Height" global. Here is a little example
of what I mean:

```
int <Project_Name>ChangeWindow( void )
{
    <Project_Name>Left   = <Project_Name>Wnd->LeftEdge;
    <Project_Name>Top    = <Project_Name>Wnd->TopEdge;
    <Project_Name>Width  = <Project_Name>Wnd->Width;
    <Project_Name>Height = <Project_Name>Wnd->Height - Wnd->BorderTop;
    return( 1 );
}
```

Or when the IDCMP handler isn't generated:

```
while( 1 ) {
    WaitPort( <Project_Name>Wnd->UserPort );
    while( imsg = GT_GetIMsg( <Project_Name>Wnd->UserPort )) {

        Class = imsg->Class;
        GT_ReplyIMsg( imsg );

        switch ( Class ) {

            case    IDCMP_CHANGEWINDOW:
                <Project_Name>Left   = <Project_Name>Wnd->LeftEdge;
                <Project_Name>Top    = <Project_Name>Wnd->TopEdge;
                <Project_Name>Width  = <Project_Name>Wnd->Width;
                <Project_Name>Height = <Project_Name>Wnd->Height -
                                       <Project_Name>Wnd->BorderTop;
```

```
                            break;

                            .....
                  }
            }
      }
```

Also the main file contains the Gadget Array's for each project window
which are define as follows:

struct Gadget *<Project_Name>Gadgets[];

To access the pointers in this array the program generates DEFINES or
EQU'S in the header file which specify the position of a certain gadget
in the array. These defines are build as follows:

#define GDX_<Gadget_Label>

Be sure that the   "gadtools.library", "intuition.library","graphics.
library" and "utility.library" are all open BEFORE any of the   generated
routines is called.   When you use a diskfont and the   "Generate
OpenFont" switch is on in the Asm or C Preferences window the "diskfont.
library" must also be opened.

For the assembler source   the valid pointers to these libraries must
be stored globally with the names "_GadToolsBase", "_IntuitionBase",
"_GfxBase", "_UtilityBase" and "_DiskfontBase" otherwise you get problems
linking.

NOTE: When you use font adaptable source which adapts to the system font
      you also need to open the "diskfont.library" and   store the
      pointer in a global called "_DiskfontBase" for the assembly source.

1.101 The IDCMP Handler

```
****************************************************************************
* Currently there are three builtin source generators. C, Assembler and *
* Oberon. The next release  will not  have  builtin  source  generators *
* anymore. By then all the  source  generators  should  be  external to *
* create a more flexible environment.                                   *
*                                                                       *
* If you study the "gadtoolsbox.library"  docs  and the  source  to the *
* sample source generator you can also build your own.                  *
****************************************************************************
```

In version 37.157++ of GadToolsBox the C source generator will generate
an IDCMP handler when wanted. This IDCMP handler is a bit experimental
and it will be changed. The generated IDCMP handler will work for most
people but some may not like the way it has been done.

As of version 37.157 GadToolsBox generates an IDCMP handler routine for
each IDCMP flag. This handler will process all incoming IDCMP messages
from the window and it will also process   all gadget and menu events.
GadToolsBox generates the following routine(s) for the event-handling:

```
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» int Handle<Project_Name>IDCMP( void );                                    »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
```
This is the main IDCMP event-handler. There will be as many of these
routines in the file as there are windows in the file you generated.
You must call this routine when you have received a signal that there
is a message pending at the window port. This routine will return the
value that the IDCMP specific routine returns. This return code can
be used to quit the program etc.

```
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
» int Handle<Project_Name>Verify( void );                                   »
»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»»
```
All windows in the generated file which has one or more of the VERIFY
flags turned on will have this routine. The handling of the VERIFY
events differs from the normal event in that the routine is called
BEFORE the message is replyed. This means that the routines for
these verify messages should _NOT_ take to long to return.

The routine labels called for by the various IDCMP messages are
constructed the following way:

```
    Flag                             Routine Name
    --------------------------------------------------------------
    IDCMP_MOUSEBUTTONS               <Project_Name>MouseButtons();
    IDCMP_MOUSEMOVE                  <Project_Name>MouseMove();
    IDCMP_CLOSEWINDOW                <Project_Name>CloseWindow();
    IDCMP_MENUVERIFY                 <Project_Name>MenuVerify();
    IDCMP_MENUHELP                   <Project_Name>MenuHelp();
    IDCMP_REQSET                     <Project_Name>ReqSet();
    IDCMP_REQCLEAR                   <Project_Name>ReqClear();
    IDCMP_REQVERIFY                  <Project_Name>ReqVerify();
    IDCMP_NEWSIZE                    <Project_Name>NewSize();
    IDCMP_SIZEVERIFY                 <Project_Name>SizeVerify();
    IDCMP_ACTIVEWINDOW               <Project_Name>ActiveWindow();
    IDCMP_INACTIVEWINDOW             <Project_Name>InActiveWindow();
    IDCMP_VANILLAKEY                 <Project_Name>VanillaKey();
    IDCMP_RAWKEY                     <Project_Name>RawKey();
    IDCMP_NEWPREFS                   <Project_Name>NewPrefs();
    IDCMP_DISKINSERTED               <Project_Name>DiskInserted();
    IDCMP_DISKREMOVED                <Project_Name>DiskRemoved();
    IDCMP_INTUITICKS                 <Project_Name>IntuiTicks();
    IDCMP_IDCMPUPDATE                <Project_Name>IDCMPUpdate();
    IDCMP_CHANGEWINDOW               <Project_Name>ChangeWindow();
```

This means that when you receive an IDCMP_MENUHELP message at your window
the IDCMP event-handler will automatically call <Project_Name>MenuHelp().

The Menu and Gadget events are processed a little differently. For each
gadget and menu a reference to a routine is generated. The gadget routine
labels are constructed as follows:

```
    <Gadget_Label>Clicked();
```

And the menu routine labels are constructed as:

```
    <Project_Name><Menu_Label>();
```

None of the routines called for by the generated event-handler will be
passed any arguments. You can get the IDCMP event specific data from a
global copy of the arrived message.   Each window in the generated file
will have a copy of the IntuiMessage that arrived at the window port. The
source generator will generate a struct IntuiMessage for each window and
the label for those IntuiMessage is constructed as follows:

    struct IntuiMessage <Project_Name>Msg;

The contents of the message that arrives at the window port will be
copied into this IntuiMessage before the IDCMP specific routine is
called. This way you can get the data you need from the IntuiMessage.

When you have the "Templates" option switched on the source generator
will also generate a file with template routines for each routine the
IDCMP event-handlers will call.   The name of the generated sources is
constructed as follows:

When you generate your source as "Source.c" the files generated will be
called:

    Source.c   -   The main source file
    Source.h   -   The source header file
    Source_temp.c -   The template source file

The way the gadget and/or menu routines are called is done by setting
the address of the routine in question in the UserData field of the
gadget/menu. Then when the IDCMP event-handler receives a gadget/menu
message the routine address is picked out of the gadget/menu structure
and then called.

This is a whole lot smaller and faster than to setup a big switch/case
table.

To keep things clean the source generator will not generate routines for
Menus, Items with subitems, BARLABEL (sub)items, TEXT gadgets and NUMBER
gadgets. These menus and gadgets never generate an IDCMP event so it's
not necessary to generate routines for them.

The generated menu-event handler supports drag-selection which is the
only official way to handle menu messages.

1.102 Font Adaptivity

When you are editing a font adaptive GUI the source generators will
create source that will automatically adapt the gadget, bevel-box and
window sizes and positions to the active font. The resulting source also
checks to see wether or not the window will fit on the screen.   If it
doesn't fit anymore the topaz 8 font will be used.

GadToolsBox will allow you to use any font when designing a font
adaptible GUI. Please be extra carefull when you design using a large
font because if you would (for example) create a SCROLLER of 10 pixels
high and at run time a smaller font is active the routine that

re- calculates the gadget could make the SCROLLER to small.

In the generated source a routine called "ComputeFont()" is generated. This routine calculates the font size and it will check if the window still fit's on the screen using the font. If the window does not fit anymore this routine will automatically switch to the topaz 8 font. This routine is always static and you should never need to call this yourself.

Two kinds of adaptable source can be generated. The first kind uses the font of the screen on which the window opens and the second will use the system default font (GfxBase->DefaultFont).

Also two routines called "ComputeX()" and "ComputeY()" are generated which re-calculate the horizontal (X) and vertical (Y) values for the gadgets, bevel-boxes, windows and texts.

NOTE: You should take special care when you place fixed-size gadgets like the CheckBox, GetFile and MX gadgets. Because they have a fixed size the gadgets are not "size-adapted" with different fonts.

1.103 Credits

I would like to thank the following people for their many suggestions and bug reports.

```
_____    T H A N K S    _____
```

Wolfgang Küting,   Kasper Peeters,   Stefan Becker,   Nico François,
Sebastiano Vigna, Andreas Jung,   Ken Simpson,   Kai Bolay,
Ola Olsson,   Carsten Feilhaber,   Richard Waspe,   Garry Glendown
Jochen Kreimer,   Georg Hessmann,   Martin Laubach,   Eugene Pang,
Klaus Nachtigall, Andreas R Kleinert, Gael Marziou,   Rene Petersen,
Magnus Holmgren,   Greg Woolley,   Marcel Offermans, Peter Kunst,
Reza Elghazi,   Steve Loughran,   Ugo Paternostro,   Franz Schwarz,
Roy v/d Woning,   Jim Hawkins,   Sven Güttner,   Martin Blom,
Lars Hecking,   Paul Hammant,   Mikael Berglund,   Ekkehard Goebel,
Stephan Fuhrmann, Andreas Blohm,   Steve Robbins,   Daniel Barrett

And special thanks to:

    Matthew Dillon   For his incredible Dice (you should register...)
    Stefan Becker   For sending me my Dice update
    Fred Fish   For the huge pile of AmigaLibDisks he produces
    Huub Schuivers   For installing 2MB on my A2630 card
    Andreas Jung   For his donations
    Ingo Sabel   For his donation
    Frederic Hahn   For the bottle of white wine
    Andreas R Kleinert   For his donation
    Klaus Nachtigall   For his donation
    Reza Elghazi   For the CD
    Sebastiano Vigna   For the font adaptivity idea
    U2   For the _great_ concert in the Westfalenhallen in
                         Dortmund. It's to bad that someone banged up my
                         car that day...   If you have a chance you really
                         should go and see U2 on their ZOO TV Tour.

1.104 Notes

At this time there really isn't much to say about the program.   It is a
tool that will (or at least should) save you a lot of time in creating
a GadTools user interface for your programs.

When a big font is used in the program it can happen that some requesters
won't fit on the screen anymore. Intuition will automatically re-size the
requester so that it will fit on the screen but some gadgets might not be
reachable anymore.

When you want to save and load crunched GUI(s) you must copy the V35+
version of the powerpacker.library in your libs: directory and you must
set-up T: as a logical device.   To do this you must have something like
this in your startup-sequence:

```
        MakeDir RAM:T
        Assign  T:  RAM:T
```

If   someone   wants to translate this manual into her/his language
permission is hereby granted. The only thing I ask is that you send me a
copy of the translation so that I can distribute it with the   next
release of the program.

Although this program itself does not follow all the rules described
in the "User Interface Style Guide" I still strongly suggest you try to
follow the rules described in that guide.

From v1.3 on GadToolsBox has an embedded version string in the "User
Interface Style Guide" format. When submitting a bug-report please tell
me the version of the GadToolsBox program you are using.   This can be
established by typing "Version GadToolsBox" in the shell. Also before
you start sending me bug-reports and suggestions please read the "TODO"
file first.

This is a program for programmers.   A lot of things in the manual might
not be clear to other Amiga users. If you plan to program the Amiga you
should at least get the RKM manuals: Libraries, Devices, Includes and
AutoDocs, User Interface Style Guide.   Also one or two good books
concerning the language you wish to learn are essential. I'm not saying
this because I like to see you throwing your money away but it's almost

impossible to program the Amiga without good documentation.

Never make changes to the source that GadToolsBox generates.   You should
not do this because each time you re-generate the source you have to make
these changes again.   You should always code your program specific
routines and stuff in one or more seperate modules.   This way you can
always add things to your GUI using GadToolsBox without having the
trouble of adding other things to the GadToolsBox generated source over
and over again.

Although GadToolsBox generates very compact source there might be some
things that you think are strange.   For example when you generate source
for a window with ten GETFILE gadgets and you look at the source the
Gadget Kind array will show GENERIC_KIND 10 times and the gadget TagItem
array will show TAG_DONE 10 times.   This is the way the source works.
Do not try to enhance it to save a couple of bytes in the resulting
executable. Please remember that it's machine generated source.   Coding
this stuff by hand might save you a few bytes on the executable size of
the resulting program but using GadToolsBox will save you hours and hours
of boring GUI coding.

Some people have asked me how they could determine the size of the window
they had created. This is really simple. Move the mouse pointer past the
right and bottom edge of the window and the coordinates will tell you the
size the window has.

Bug reports, suggestions, postcards, flames, criticism, contributions,
ideas, gifts etc.. to:

Jan van den Baard   Fido: 2:285/502.7 (Jan van.den.Baard)
Bakkerstraat 176
3082 HE Rotterdam
Holland

     »» "Trust me. I know what I'm doing."   -: Sledge Hammer ««

     »» "Life sucks."   -: Al Bundy   ««

     »» "Hasta la vista, Baby."   -: The Terminator ««

     »» "Make noise, Tell the world to fuck off!" -: Bono   ««

     »» "The most mercifull thing in the world, I"
        "think, is the inability of the human mind"
        "to correlate all it's contents.   " -: Propaganda   ««

     »» "Normal is boring."   -: Fido Dido   ««