

# **InterfaceEditor**

Simone Tellini

COLLABORATORS
---------------

	TITLE : InterfaceEditor		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Simone Tellini	August 22, 2024	

REVISION HISTORY
------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>InterfaceEditor</b>	<b>1</b>
1.1	Interface Editor User Manual . . . . .	1
1.2	Disclaimer . . . . .	1
1.3	Copyright and Distribution . . . . .	2
1.4	How to register . . . . .	3
1.5	Updates . . . . .	3
1.6	Introduction . . . . .	3
1.7	Minimal configuration required . . . . .	4
1.8	Starting with Interface Editor . . . . .	5
1.9	Making a Gadget . . . . .	5
1.10	Gadget Selection . . . . .	6
1.11	Moving & Sizing Gadgets . . . . .	6
1.12	Gadgets/Size... . . . .	7
1.13	Gadgets/Copy . . . . .	7
1.14	IEditor's Menus . . . . .	7
1.15	Project/New . . . . .	9
1.16	Project/Open... . . . .	9
1.17	Project/Save . . . . .	10
1.18	Project/Save As... . . . .	10
1.19	Project/Generate Source... . . . .	10
1.20	Preferences/Workbench . . . . .	10
1.21	Project/About... . . . .	10
1.22	Project/Exit . . . . .	10
1.23	Gadgets/Remove . . . . .	11
1.24	Gadgets/Tags... . . . .	11
1.25	Gadgets/Align/Left . . . . .	12
1.26	Gadgets/Align/Right . . . . .	12
1.27	Gadgets/Align/Top . . . . .	12
1.28	Gadgets/Align/Bottom . . . . .	12
1.29	Gadgets/Clone/Width . . . . .	13

---

1.30	Gadgets/Clone/Height . . . . .	13
1.31	Gadgets/Clone/Both . . . . .	13
1.32	Gadgets/Load... . . . .	13
1.33	Gadgets/Save... . . . .	13
1.34	Gadgets/Items... . . . .	13
1.35	Gadgets/Font... . . . .	14
1.36	Gadgets/Select all . . . . .	14
1.37	Window/New... . . . .	14
1.38	Window/Discard . . . . .	14
1.39	Window/Discard All . . . . .	14
1.40	Window/Open... . . . .	14
1.41	Window/Load... . . . .	14
1.42	Window/Save... . . . .	14
1.43	Window/Flags... . . . .	15
1.44	Window/IDCMP... . . . .	15
1.45	Window/Tags... . . . .	15
1.46	Windows/Close . . . . .	16
1.47	Windows/Close All . . . . .	16
1.48	Windows/Title... . . . .	16
1.49	Windows/Size... . . . .	16
1.50	Windows/Zoom... . . . .	17
1.51	Screen/Font... . . . .	17
1.52	Screen/DriPens... . . . .	17
1.53	Screen/Palette/Modify... . . . .	18
1.54	Screen/Palette/Load... . . . .	18
1.55	Screen/Palette/Save... . . . .	19
1.56	Screen/Tags... . . . .	19
1.57	Screen/Type... . . . .	20
1.58	Screen/Save... . . . .	20
1.59	Screen/Load... . . . .	20
1.60	Misc/Localization... . . . .	21
1.61	Gadget Tags . . . . .	22
1.62	BUTTON . . . . .	22
1.63	CHECKBOX . . . . .	23
1.64	INTEGER . . . . .	23
1.65	LISTVIEW . . . . .	24
1.66	MX . . . . .	25
1.67	NUMBER . . . . .	25
1.68	CYCLE . . . . .	26

---

1.69 PALETTE . . . . .	26
1.70 SCROLLER . . . . .	27
1.71 SLIDER . . . . .	27
1.72 STRING . . . . .	28
1.73 TEXT . . . . .	29
1.74 Macros/Function keys... . . . .	30
1.75 Macros/Add... . . . .	30
1.76 Macros/Remove... . . . .	30
1.77 Macros/Execute... . . . .	30
1.78 ARexx scripts shipped with IEditor . . . . .	31
1.79 Preferences/Tools Window . . . . .	32
1.80 Preferences/Use Gadgets . . . . .	32
1.81 Preferences/Current window in front of . . . . .	32
1.82 Preferences/Use WFlags . . . . .	32
1.83 Preferences/Generator... . . . .	32
1.84 Preferences/Asm Sources... . . . .	32
1.85 Preferences/C Sources... . . . .	33
1.86 Preferences/Save . . . . .	34
1.87 Example of a headers definition file . . . . .	34
1.88 Font Adaptivity . . . . .	35
1.89 The IDCMP Handler . . . . .	36
1.90 The Gadget Activation Key Handler . . . . .	36
1.91 The Generated Source . . . . .	36
1.92 ARexx Routines . . . . .	39
1.93 Main Routines . . . . .	41
1.94 Gadget banks handling . . . . .	43
1.95 MultiSelect Listviews handling . . . . .	43
1.96 Gauge.iex . . . . .	44
1.97 Credits . . . . .	45
1.98 Notes . . . . .	45
1.99 Preferences/Create Icons . . . . .	47
1.100 Images bank . . . . .	47
1.101 Boolean Gadgets . . . . .	47
1.102 Future plans . . . . .	49
1.103 Keyboard commands . . . . .	49
1.104 Windows/Print . . . . .	50
1.105 Gadgets/Spread/Horizontally . . . . .	50
1.106 Gadgets/Spread/Vertically . . . . .	50
1.107 Gadgets/Spacing/Set X... . . . .	50

---

1.108Gadgets/Spacing/Set Y...	51
1.109Windows/BevelBoxes/Add	51
1.110Windows/BevelBoxes/Edit...	51
1.111Windows/Images/Add	51
1.112Windows/Images/Move	51
1.113Windows/Images/Delete	52
1.114Windows/Texts/Add...	52
1.115Windows/Texts/Delete...	52
1.116Windows/Texts/Edit...	52
1.117Windows/Texts/Move...	53
1.118Windows/Gadget bank/Create...	53
1.119Windows/Gadget bank/Delete...	53
1.120Windows/Gadget bank/Show...	53
1.121Windows/Gadget bank/Hide...	53
1.122Windows/Gadget bank/Parameters...	54
1.123Menu Editor	54
1.124ARexx Interface Editor	55
1.125Misc/main()...	56
1.126IEditor's author... err...hey, it's me! :)	57
1.127The Tools Window	57
1.128Gadgets/TabCycle order...	58
1.129Project/Parameters...	58
1.130Shared IDCMP Port	59
1.131ARexx commands	60
1.132QUIT	61
1.133OPEN	61
1.134SAVE	61
1.135SAVEAS	61
1.136GETNAME	62
1.137SETNAME	62
1.138NEW	63
1.139GETACTWDATA	63
1.140GETWNDDATA	64
1.141GENERATEASM	64
1.142GENERATEC	64
1.143GENERATEE	64
1.144GETIMAGE	64
1.145GETIMAGEATTR	65
1.146SETIMAGEATTR	65

1.147ADDBOX . . . . .	66
1.148SETBOXATTR . . . . .	66
1.149ADDITEXT . . . . .	66
1.150GETITEXT . . . . .	67
1.151GETITEXTATTR . . . . .	67
1.152GETITEXTLENGTH . . . . .	67
1.153SETITEXTATTR . . . . .	68
1.154GETDRIPEN . . . . .	68
1.155GETBOX . . . . .	68
1.156GETBOXATTR . . . . .	68
1.157GETFONTATTR . . . . .	69
1.158GETSCRFONT . . . . .	69
1.159GETFILE . . . . .	69
1.160Index . . . . .	70

## Chapter 1

# InterfaceEditor

### 1.1 Interface Editor User Manual

Interface Editor - User Manual

```
~Disclaimer~~~~~
~Gadget~Selection~~~~~
~Copyright~~~~~
~IEditor's~Menus~~~~~
~Updates~~~~~
~Gadget~Tags~~~~~
~How~to~register~~~~~
~Moving & Sizing~gadgets  ~~
~Requirements~~~~~
~The~Generated~Source~~~~~
~Introduction~~~~~
The ARexx interface
~Starting~IEditor~~~~~
~Notes~~~~~
Keyboard commands
~Credits~~~~~
The ToolBar
~A~look~to~the~future~~~~~
```

History

Search...

### 1.2 Disclaimer

THERE IS NO WARRANTY FOR THE SOFTWARE TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHERE OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL



ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE SOFTWARE AS PERMITTED BELOW, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE SOFTWARE TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 1.3 Copyright and Distribution

Interface Editor

© Copyright 1994-96 Simone Tellini Software

Written using DICE C v2 by  
Simone Tellini  
Piazza Resistenza 2  
42016 Guastalla RE  
Italia

Fidonet: 2:332/502.18 (Simone Tellini)  
Internet: wiz@pragmanet.it

User Manual

© Copyright 1995-1996 Simone Tellini Software

regtools.library is © Copyright Nico François

Interface Editor is (C) 1994-96 by Simone Tellini Software. This program is FREEWARE.

This program may be freely distributed provided that 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 documentation of the program. I would appreciate also to receive a copy of that program, or, at least, a demo! :-)

Commercial distribution of Interface Editor and/or its relative files is \*NOT\* allowed without written permission from the author. This also means the distribution on cover disks, disk magazines etc.

I'll be glad to give the permission to insert this program on disk magazines or cover disks if I receive a copy of it. Write me for details.

Only the output of IEditor is not subjected to these rules.

---

## 1.4 How to register

IEditor is now freeware. Anyway, if you think that it or its sources helped you, you can always send me a contribution. It's up to you.

## 1.5 Updates

Once registered, you'll receive a special keyfile which will let you use any copy of IE without limitations. So, you can get updates of IE directly from the PD distribution networks.

You can always find the latest version of IEditor on Aminet, in the 'dev/gui' directory.

Every registered user who has got an E-Mail address will receive a message when a new version of IE is released.

IEditor is also present on the World Wide Web. Its URL is the following:

<http://www.pragmanet.it/~tellini/main.html>

Currently this BBS supports IEditor:

- BBS:	The Drake
SysOp:	Paolo Masetti
Tel:	+39-522-838800
Fido node:	2:332/522
Message Area:	IEDITOR.LOC
File name:	Area SUPPORT - IEditxxx.lha (e.g. IEdit229.lha)
Magic File Name:	IEDITOR

If some other BBS could support IEditor, I'll be very glad: write me for details.

## 1.6 Introduction

Interface Editor lets you save a huge amount of time when you have to design a GadTools User Interface for your programs. This program's the successor of Interface Editor v1.0, which was a little program with very little control on the aspect of the interface. Starting with version 2.0, I've tried to make IEditor a good user-friendly program; I've also improved the output that, in v1.0, was very bad!

These are the main features of IEditor v2:

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

Full control over the window flags and tags

Fully mouse and keyboard controlled

Multiple windows

User selectable font for every gadget

The possibility to change screen resolution while editing

Easy gadget alignment and dimensioning

Multiple gadget moves, deletes, edits and copies.

Boolean gadgets, Bevel Boxes, MENU\_IMAGES

ARexx Port

ARexx Interface Editor

Expandible using external modules

...

## 1.7 Minimal configuration required

InterfaceEditor requires Kickstart 2.04+ and 512kb RAM at least.

It's advisable to use at least a 8 kbytes stack for IE.

These libraries are used by IEditor:

regtools.library	v37+
icon.library	v37+
locale.library	v38+ (optional)
asl.library	v37+
gadtools.library	v37+
intuition.library	v37+
dos.library	v37+
graphics.library	v37+
diskfont.library	v37+
iffparse.library	v37+
rexsyslib.library	(optional)

To compile the output of IEditor you need a C compiler (like DICE) or an assembler with 2.0 includes or the Amiga E compiler (EC) v2.1b or better.

If you generate the Assembly sources with the RawCode parameter, the include files won't be needed.

InterfaceEditor has been tested on these machines:

---

- Amiga 1200 68EC020 28MHz  
2Mb CHIP 4Mb FAST Ram ( Blizzard 1220/4 )  
HD IDE Conner 1.2 Gb  
Squirrel Classic  
CD-ROM SCSI Pioneer DR-U124X  
2 Floppy Drives  
Kick v39.106 - WB v39.29 ( Release 3.0 )
- Amiga 4000/40 68040 25MHz  
2Mb CHIP 16Mb FAST Ram  
HD IDE Conner 210 Mb  
HD IDE Seagate ST144At 124 Mb  
Kick v39.106 - WB v40.42 ( Release 3.0 )
- Amiga 2000B 68000 7MHz  
1Mb CHIP 2Mb FAST Ram  
HD SCSI 850 Mb  
Kick v40.63 - WB v40.42 ( Release 3.1 )

## 1.8 Starting with Interface Editor

You can run Interface Editor from either the Shell or the Workbench. From Shell, it accepts as a argument the name of a previously saved GUI file. From WB, it accept as WB argument the icon of a previously saved GUI... ;)

When it's loaded, you will see a screen with a little window: the ToolBar. Then you can load an old interface or start creating a new one.

To load a GUI you can select the item Open... in the first menu or you can click on the 10th gadget of the ToolBar and then select the file you want to load.

To start creating a new interface, you must click on the first gadget of the ToolWindow or select the item New... in the "Windows" menu.

## 1.9 Making a Gadget

Making a gadget is very simple. First of all you have to select the item "Add..." in the "Gadgets" menu or click on the 5th gadget of the ToolBar. Then you must choose the kind of the gadget in the listview which will appear.

Now move the mouse pointer to where you want the top-left corner of the gadget to be (you'll see the coordinates on the ScreenBar). Click LMB (left mouse button) and move the mouse while olding down the button. You'll see a box which represents your gadget, whose dimensions are showed in the ScreenBar.

Once you've decided the size of the gadget, release the mouse button.

A requester will pop up: here you can edit the gadget title. After that another request will pop up letting you modify the gadget tags.

---

Then, the gadget will be created. If the creation operation fails, you'll see a black rectangle instead of the gadget.

Here's a list of the gadget supported by IE2:

```
~BUTTON~~~~~  
~CHECKBOX~~~~~  
~INTEGER~~~~~  
~LISTVIEW~~~~~  
~MX~~~~~  
~NUMBER~~~~~  
~CYCLE~~~~~  
~PALETTE~~~~~  
~SCROLLER~~~~~  
~SLIDER~~~~~  
~STRING~~~~~  
~TEXT~~~~~  
~BOOLEAN~~~~~
```

Note that the CHECKBOX and MX gadgets have a fixed size which means that you cannot make them any size you wish (unless you set the scaled tag of the gadget). Moreover, with placing these gadgets, there can be some problems if you use a font bigger than Topaz 8 and don't set the scaled tag.

NOTE: the BOOLEAN gadget type isn't handled by gadtools.library, but it's a simple intuition gadget.

## 1.10 Gadget Selection

If the gadgets aren't disabled or detached, to select one of them you have to press LMB where there are no gadgets, then (holding down the mouse button) draw a rectangle so that the gadget(s) that you want to select is/are partly or completely covered by the box.

If the gadgets are disabled or detached you can simply click on the gadget you like to select. You can always drag a box around more than one gadget, to do a multiple selection.

Moreover, if you hold down the SHIFT key, the gadgets currently active won't be deactivated, but will remain active.

When a gadget is activated, it will have a box around it with four dot in the corners.

## 1.11 Moving & Sizing Gadgets

It is possible to move or resize one or more gadgets at the same time, depending on how many gadgets are selected.

---

To move a gadget, simply click on it and move the mouse holding down the mouse button. (If you want to move more than a gadget at the same time, you'll have to keep a Shift key pressed)

You'll see boxes representing your gadgets, when they're where you want, just release the mouse button. It wasn't so hard, was it?

To size a gadget you'll have to click in one of its four corners; then move the mouse while holding down the button. Release the mouse button to perform the changes.

## 1.12 Gadgets/Size...

You can choose to size a gadget (or more than one) by specifying its width and height or by sizing them with your mouse.

In the first case, select the item "Gadgets/Size...": a requester will pop up, enter here the values you want (also X and Y coordinates can be entered here).

## 1.13 Gadgets/Copy

This will create a copy of the selected gadgets: after selecting this item, you will see boxes representing the selected gadgets that will follow the mouse pointer. When they're where you want them to be, press LMB.

The new gadgets will be created in that place and will have ALL the attributes equal to the original gadgets, labels included.

## 1.14 IEditor's Menus

Well, here you find the description of all the menus of my program:

```
Project
~About...~~~~~  ~~
~New~~~~~      ~~~~
~Open...      ~~~~~~
~Save~~~~~    ~ ~
~Save~As...~~ ~ ~~~
Parameters...
Generate Source...
~Exit...~~~~~
```

```
Windows
~New...~~~~~
~Open...~~~~~
~Close~~~~~
~Close~All~~~~~
~Discard~~~~~
```

---

```
~Discard~All~~~~~
~Title...~~~~~
~Flags...~~~~~
~IDCMP...~~~~~
~Size...~~~~~
~Zoom...~~~~~
~Tags...~~~~~
  BevelBoxes
    Add
    Edit...
  Images
    Add
    Move
    Delete
  Texts
    Add...
    Delete...
    Edit...
    Move...
  Gadget Banks
    Make...
    Delete...
    Show...
    Hide...
    Parameters...
  Print
~Save...~~~~~
~Load...~~~~~
```

## Gadgets

```
~Add...~~~~~
~Remove~~~~~
~Select~All~~~~~
~Items...~~~~~
  Size...
~Tags~~~~~
~Font~~~~~
  Align
    ~Right~~~~~
    ~Left~~~~~
    ~Top~~~~~
    ~Bottom~~~~~
  Spread
    Horizontally
    Vertically
  Spacing
    Set X...
    Set Y...
  Clone
    ~Width~~~~~
    ~Height~~~~~
    ~Both~~~~~
~Copy~~~~~
  TabCycle order
~Save...~~~~~
~Load...~~~~~
```

## Screen

```

~Tags...~~~~~ ~
~Type...~~~~~ ~
~Font...~~~~~ ~
~DriPens...~ ~~~~~
  Palette
    ~Modify~~~~~ ~
    ~Load...~~~~~ ~
    ~Save...~~~~~ ~
~Save...~~~~~ ~
~Load...~~~~~ ~

```

## Misc

```

~Menu~Editor...~~
~Image~Bank...~~~
  ARexx Editor...
  main()...
  Localization...

```

## Macros

```

  Function keys...
  Add...
  Remove...
  Execute...

```

## Preferences

```

~Tools~Window~~~~~
~Use~Gadgets~~~~~
~Workbench...~~~~~
~Current~Window~In~Front~
~Use~WFlags~~~~~
~Asm~Source...~~~~~
~Generator...~~~~~
~Create~icons~~~~~
~Save~~~~~

```

## 1.15 Project/New

This will remove from memory the current GUI, so that you can start creating a new one from scratch.

If the GUI hasn't be saved, you will be asked if you want to continue or to abort the operation.

## 1.16 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) Yes      -    Delete the GUI in memory.
- B) No       -    Abort the operation.



When the loading operation is finished you will be able to select a window to open with the menu Windows/Open...

## 1.17 Project/Save

This will save the GUI currently in memory to the name under which it was saved last time. If it's the first time you save it, a file requester will pop up asking for a file name.

## 1.18 Project/Save As...

This has the same function of "Project/Save...", but always opens the file requester.

## 1.19 Project/Generate Source...

This will present you with a filerequester in which you must select the name which will be used to save the generated source.

If you haven't chose a generator yet, its configuration window will pop up. Otherwise, IE will use the generator you've chosen before.

Please refer to ~The~Generated~Source~ chapter for more information.

## 1.20 Preferences/Workbench

This will try to close the Workbench (or open it if it was closed). If you close the WB and then exit from IEditor, the Workbench will be automatically opened again.

## 1.21 Project/About...

I think you know what it does, don't you? Well, it opens that window with copyright infos and my address that you should see every time you load Interface Editor (unless you are a registered user ;).

## 1.22 Project/Exit

Selecting this item, you will leave IEditor. If the current GUI is not saved, IE will ask you if you want to "Save", save it in binary format, "Exit", quit immediatly or "Cancel", don't quit.

---

## 1.23 Gadgets/Remove

Selecting this item, a requester will ask you if you really want to delete the selected gadget. Answering yes to it, will cause IEditor to delete them, for ever.

You can delete the gadgets also by pressing the 'DEL' key.

REMEMBER: Once a gadget is deleted, you cannot bring it back!

## 1.24 Gadgets/Tags...

This will open a requester to edit the tags of the gadget(s) selected.

For every gadtools gadget, you'll find these gadgets:

Title:	Insert here the gadget text.
Label:	This will identify the gadget in the source. Note that if you don't specify it, it will be created when the source is generated.
Title Position:	This lets you select where the gadget text should be placed.
Underscore:	Check it if you want an underlined letter in the gadget text. The letter to underline must be preceded by "_".
Highlight Title:	Check it if you want the title to be written with the Highlight Pen of the screen. With some gadgets kind, it has no effect.

Other gadgets are present according to the kind of the gadget.

Select the kind you want to know more about:

```
~BUTTON~~~~~
~CHECKBOX~~~~~
~INTEGER~~~~~
~LISTVIEW~~~~~
~MX~~~~~
~NUMBER~~~~~
~CYCLE~~~~~
~PALETTE~~~~~
~SCROLLER~~~~~
~SLIDER~~~~~
~STRING~~~~~
~TEXT~~~~~
```

This gadget type is not handled by gadtools.library, so it hasn't got any tag, but maybe you want to know which parameters you can modify... ;)

~BOOLEAN~~~~~

## 1.25 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~ gadgets will inherit the left-edge coordinate of the clicked gadget.

## 1.26 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~ gadgets will inherit the right-edge coordinate of the clicked gadget.

NOTE: This might not always work when you align to the right-edge of a ~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.

## 1.27 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~ gadgets will inherit the top-edge coordinate of the clicked gadget.

NOTE: This might not work when you try to align a ~MX~ gadget, if this gadget uses a font higher than 8 and doesn't have the scaled tag set to true.

## 1.28 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~ gadgets will inherit the bottom-edge coordinate of the clicked gadget.

NOTE: This might not always work when you align to the bottom-edge of a ~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.

NOTE 2: This might not work when you try to align a ~MX~ gadget, if this gadget uses a font higher than 8 and doesn't have the scaled tag set to true.

---

## 1.29 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~ gadgets will get the same width as the clicked gadget.

## 1.30 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~ gadgets will get the same height as the clicked gadget.

## 1.31 Gadgets/Clone/Both

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~ gadgets will get the same dimensions as the clicked gadget.

## 1.32 Gadgets/Load...

This will open an ASL filerequester with which you can select a set of previously saved gadgets. Those gadgets will be placed in the active window and they don't replace the gadgets already existing in it.

## 1.33 Gadgets/Save...

This will save all ~selected~ gadgets.

## 1.34 Gadgets/Items...

This will open the Items Editor: it consists of a listview in which you can see the items that the selected gadget has, a string gadget which lets you modify the items and three buttons:

New:	Adds a new item to the list.
Delete:	Delete the selected item.
Ok:	(=ESC) Close the editor.

This function works only with LISTVIEW, MX and CYCLE gadgets. For the last two ones, you are obliged to insert at least TWO items.

---

### 1.35 Gadgets/Font...

This will open an ASL Font Requester in which you can select a font. This font will be used for all the selected gadget.

### 1.36 Gadgets/Select all

Yeah! You guessed it! This will select all the gadgets of the current window. ;)

### 1.37 Window/New...

This will open a new blank window.

You can open as many window as you want (you must have enough memory, of course).

### 1.38 Window/Discard

The currently active window will be deleted from memory. BE CAREFUL!

### 1.39 Window/Discard All

All windows will be deleted from memory. BE VERY CAREFUL!

### 1.40 Window/Open...

A list of the currently in memory windows will appear: click on the one you want to open. You can use also the arrow keys to go up and down the list, the return key to select and the ESC key to abort.

### 1.41 Window/Load...

This will open an ASL filereq with which you can choose a window to load. The selected window will be appended to the other windows of the interface.

### 1.42 Window/Save...

This opens a filereq in which you can select the name to save the active window to.

---

### 1.43 Window/Flags...

This will pop up a list of the windows flags: you can choose the ones you wish by clicking on them. The selected ones will be marked by this '\*'.

Note that the flags aren't set, unless you have checked the menu item ~Preferences/Use~WFlags~.

### 1.44 Window/IDCMP...

This will pop up a list of the windows IDCMP: you can choose the ones you wish by clicking on them. The selected ones will be marked by this '\*'.

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

### 1.45 Window/Tags...

This will open up a window in which you can edit the most important window tags.

In the window you will find the following gadgets:

ScreenTitle	You can type an alternate screen title in here. This will be the title of the screen when the window is active.
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.
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.
Notify Depth	If this tag is set to true, you will receive an IDCMP_CHANGEWINDOW when your window is depth-arranged.
Menu Help	If checked, enables the MenuHelp feature for this window.
Tablet Messages	It requests extended IntuiMessages for your window, so you can receive IESUBCLASS_NEWTABLET input

events with IntuiMessages.

Shared IDCMP Port    If checked, this window will use the shared IDCMP port.

In the Localize box, you'll find some checkboxes that will let you decide what to localize in that window.

## 1.46 Windows/Close

This will close the active window, letting you save the memory allocated for it and for its gadgets.

If you want to open it again, select the item Open... from this menu.

## 1.47 Windows/Close All

This will close all the windows of your GUI, letting you save the memory allocated for them and for their gadgets.

If you want to open them again, select the item Open... from this menu.

## 1.48 Windows/Title...

This will open the window title requester, in which you can specify the window title (Noo! ;) and the label for the window.

Note that if you don't specify a label, it will be created by IE when you generate your source.

## 1.49 Windows/Size...

This will open a requester in which you can specify all the size tags of the windows, that are:

MinWidth, MinHeight, MaxWidth, MaxHeight: these determine the minimal and maximum dimensions of the window. Pressing the "\*" gadget next to the integer gadgets, the actual dimension of the active window will be copied into them.

Inner Width, Inner Height: if selected, these will determine the inner size of the window, no care of the size of the border or the title bar.

The Ok and Cancel gadgets confirm or abort the operation respectively.

---

## 1.50 Windows/Zoom...

This will open a requester in which you'll find these gadgets:

- Top: Y coordinate of the top left corner of the window when the zoom gadget is pressed.  
-1 indicates that the window must remain where is. (Kick 3.0)
- Left: X coordinate of the top left corner of the window when the zoom gadget is pressed.  
-1 indicates that the window must remain where is. (Kick 3.0)
- Width: It indicates the width that the window should have when the zoom gadget is pressed.
- Height: It indicates the height that the window should have when the zoom gadget is pressed.
- Ok: It confirms the changes.
- Cancel: It abort the operation.

There are also four "\*" gadget: pressing them the actual size and position of the window will be copied into the respective integer gadgets.

## 1.51 Screen/Font...

This will open the ASL Font Req in which you can select a new font for the screen.

Note that this will affect the source code only if the generation of the screen has been activated.

## 1.52 Screen/DriPens...

This will open a requester in which you can edit the DriPens, that are an array of pen numbers which are used to create the OS 2.0 new look.

In this req you will find these gadgets:

- 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.

## 1.53 Screen/Palette/Modify...

This will open a standard ReqTools color requester, that will let you change the current palette of the screen.

## 1.54 Screen/Palette/Load...

This will open an ASL file requester in which you can select an IFF-ILBM file. The palette of that file will be used for the screen.

## 1.55 Screen/Palette/Save...

This will open an ASL file requester in which you can enter the name of the palette file. IEditor saves the color map in a standard IFF-ILBM file.

## 1.56 Screen/Tags...

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

Left, Top	These specify the position of the top left corner of the screen.
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.
Show Title	If it isn't checked, the screen will not have the title-bar.
Behind	Setting this, the screen will be opened behind the other ones. This lets you to prepare it before putting it on front.
Quiet	Tells Intuition not to render system screen gadgets or title.
Type	<p>This gadget will allow you to change the screen type you are creating. The following types are supported:</p> <p>CUSTOMSCREEN The screen of your GUI cannot be used by other programs.</p> <p>PUBLICSCREEN The screen of your GUI can be used by any other program.</p>
Overscan	If this is checked, in the generated source there won't be the definition of the screen size via SA_Width & SA_Height, but SA_Overscan will be used instead.
Pub Name	If you enter here a string, the screen will be a Public Screen, whose name is it.

---

Full Palette	If set, when the screen is opened, all its color map entries will be overridden by the set of preferences colors.
Error Code	If set, in the source will be generate a variable in which an extended error code will be put in the case that OpenScreenTagList fail.
Draggable	If not set, the screen won't be draggable with the title bar. This works only with kick v39+.
Exclusive	Set it if you never want your screen to share the display with another one. This means that the screen can't be pulled down and will not appear behind other screens that are pulled down. This works only with kick v39+.
Share Pens	If set, Intuition will obtain the pens for your screen in shared mode. This works only with kick v39+.
Interleaved	Requests Intuition to alloc an interleaved bitmap for your screen. This works only with kick v39+.
Like Workbench	Set it if you want a screen just like the Workbench one. This works only with kick v39+.
MinimizeISG	It tells Intuition to use the smallest Inter Screen Gap possible for your screen. This works only with kick v40+.

## 1.57 Screen/Type...

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

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.

## 1.58 Screen/Save...

This will open an ASL File Requester in which you can enter the name of the file in which you want to save the current screen definition.

## 1.59 Screen/Load...

This will open an ASL File Requester in which you can select a file to load the screen definition from.

## 1.60 Misc/Localization...

This will open a requester in which you'll find these gadget:

Catalog	name ~of ~~the ~~catalog ~~of ~~your ~~application (IE will append ".catalog")
Join	Insert here the path of a .cd file to join to ~the one generated by IE. In that way, you ~can ~insert into the catalog more strings than the ~ones ~used by your GUI. The specified file will be prepended ~to ~the ~new .cd file, so that you needn't ~to ~recompile ~all your application when you add or remove ~a ~string from ~the ~~GUI. ~Of ~course, ~you ~will ~have ~to recompile your GUI whenever you add ~or ~remove ~a string from the .cd (and .ct) file generated.  NOTE: as of version 2.31 you are encouraged not to use this field anymore, but to import the strings you need with the "Import..." gadget.
Built In	Insert here the default language of your program.
Version	Minimum catalog version required by your program. If 0, any version will be accepted.
Activated	This actives the localization of your GUI.
Languages	Here you can see ~the ~languages ~which ~IE ~will create the locale files for.
New	Adds a new language. You should specify the ~name of the language and the name of ~the ~translation file that will be created for that language. Note that the ".ct" suffix is automatically added by IEditor.
Delete	This will remove the selected ~language ~and ~ALL the translation strings related to it.
Strings	Here you can ~handle ~all ~the ~strings ~of ~your application. ~~So ~~you ~~can ~handle ~the ~whole localization process from inside IEditor. Double-clicking on a string brings up the ~string requester explained below.

---

New	Adds a new string to the database.
Delete	Remove a string from ~the ~database. ~You ~cannot delete a strings used by the GUI, of course.
Import...	This will let you scan a .cd file and import ~all the strings it contains into IEditor.

These are the gadgets of the string requester:

String	The string you're editing. Note that here you can only modify the strings ~not ~used ~by ~the ~GUI: changing a string used in the GUI won't have any result.
ID	Label for that string. You can then refer to that string using "CatCompArray[ ID ].cca_Str" in your code. The labels for the strings used in ~the ~GUI ~are assigned by IEditor when generating the source.
Translations	Translations for the string.

## 1.61 Gadget Tags

Select the kind you would like to know more about:

```
~BUTTON~~~
~CHECKBOX~
~INTEGER~~
~LISTVIEW ~
~MX~~~~~
~NUMBER~~~
~CYCLE~~~ ~
~PALETTE~ ~
~SCROLLER~
~SLIDER~~ ~
~STRING~~~
~TEXT~~~~~
~BOOLEAN~~~~~
```

## 1.62 BUTTON

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Immediate	When this tag is set, the gadget will send you an IDCMP_GADGETDOWN message when it is pressed. This works only with kick v39+.

## 1.63 CHECKBOX

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Checked	By default the created gadget is off. When you turn this option on the gadget will be created "checked".
Scaled	<p>This tags works only with Kickstart 3.0+.</p> <p>If you set it, you will be able to change the size of the checkbox gadget.</p> <p>NOTE: if you use a font higher than 8 points, the gadget won't be placed where you draw it, but a little further down. Setting this tag, it will be placed exactly.</p>

## 1.64 INTEGER

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Tab Cycle	If this tag is set, when the user presses the TAB key while the gadget is active, the next gadget will be activated.
Immediate	<p>When this tag is set, the gadget will send you an IDCMP_GADGETDOWN message when it is pressed.</p> <p>This works only with kick v39+.</p>
Exit Help	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.
Replace Mode	This kind of gadget is normally created in the insert mode. Setting this tag it will be created in the replace mode, that means that when the user presses a key, it will overwrite the character below the cursor.
Number	Normally the gadget is created with 0 as its initial value. Here you can enter the value that you want it to default to.
MaxChars	In this gadget you can set the maximum number of digits that can be entered in this gadget.

---

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

## 1.65 LISTVIEW

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled. With this gadget, this tag works only with kick v39+.
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.
Show Selected	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 will be displayed with a different background color.
Top	Top item that should be visible in the listview.
Make Visible	Number of the item that should be forced within the visible area of the listview by doing minimal scrolling. This tag overrides "Top" and works only with kick v39+.
MultiSelect	This will turn the listview into a multiselectable one. This feature works only with kick v39+. See ~this ~section ~to ~know ~how ~to ~use ~these listviews.
Scroll Width	In this gadget you can enter the width of the ~SCROLLER~ gadget in pixels.
Max Pen	The maximum pen number used by rendering in a custom rendering callback hook. This works only with kick v39+.
Selected	Number of the item that should be selected.
Spacing	In this gadget you can enter the number of pixels that is used as spacing between the list names.
Item Height	Enter here the exact height of an item. This is normally useful for listviews that use the GTLV_CallBack rendering hook. This works only with kick v39+.

---

## 1.66 MX

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled. With this gadget, this tag works only with kick v39+.
Scaled	This tags works only with Kickstart 3.0+. If you set it, you will be able to change the size of the mx gadget. NOTE: if you use a font higher than 8 points, the gadget won't be placed where you draw it, but a little further down. Setting this tag, it will be placed exactly.
Title Place	It indicates where the title should appear.
Active	This gadget enables you to set the option that is active when the gadget is created.
Spacing	In this gadget you can enter the number of pixels that is used as spacing between the options.

## 1.67 NUMBER

This are the tags you can specify with IE:

Number	By default the gadget displays 0. In here you can change the value that the created gadget will default to.
Border	When checked the created gadget will have a recessed border drawn around it.
Front Pen	The pen to use when rendering the number. -1 tells IE not to use this tag in the source. This works only with kick v39+.
Back Pen	The pen to use when rendering the background of the number. -1 tells IE not to use this tag in the source. This works only with kick v39+.
Justification	Determines where to render the number within the gadget box. This works only with kick v39+. Under v39, you must also set the tag Clipped, if you select this one, otherwise the text won't be printed.
Format	C-Style formatting string to apply on the number before display. Be sure to use the 'l' (long)

---



	modifier. This works only with kick v39+.
Clipped	If it's set, the text of the gadget will be clipped to gadgets dimensions. This works only with kick v39+.
Max Number Len	This indicates the maximum number of bytes that can be generated by applying the "Format" string.

## 1.68 CYCLE

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Active	This gadget enables you to set the option that is active when the gadget is created.

## 1.69 PALETTE

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Color	In this gadget you can enter the color that is active in the created gadget.
Depth	The depth allows you to control the amount of colors that are displayed in the created gadget.  <div style="margin-left: 40px;"> 1    = 2 colors  2    = 4 colors  .....  8    = 256 colors </div>

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

Color 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.

Indicator Width	This tag indicates the desired width of the current-color indicator. If it's equal to 0, the indicator won't be created. NOTE: This tag doesn't work with kick v39+.
Indicator Height	This tag indicates the desired height of the current-color indicator. If it's equal to 0, the indicator won't be created. NOTE: This tag doesn't work with kick v39+.
Num Colors	Number of colors to display in the palette gadget. This overrides "Depth" and allows numbers which aren't powers of 2. This works only with kick v39+.

## 1.70 SCROLLER

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Immediate	When this option is on your window will receive a message at the moment the user clicks on the gadget.
Rel Verify	When checked your window will receive a message when the user releases the left mouse button while the pointer is still over the gadget.
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.
Freedom	With this gadget you can set the direction in which the scroller is adjustable.

## 1.71 SLIDER

This are the tags you can specify with IE:

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

---

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.
Immediate	When this option is on your window will receive a message at the moment the user clicks on the gadget.
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.
Max Level Len	Maximum length in characters of level string when rendered beside the slider. This works only with kick v39+.
Level Format	In here you can enter a C-style formatting string for the level indicator. When you keep this empty there will not be a level indicator.
Level Place	With this gadget you determine the position at which the level indicator is rendered.
Max Pixel Len	It indicates the maximum pixel size used up by the level display for any value of the slider. This is useful when dealing with proportional fonts. This works only with kick v39+.
Justification	Determines how the level display is to be justified within its allotted space. This works only with kick v39+.
Freedom	With this gadget you can set the direction in which the slider is adjustable.

## 1.72 STRING

This are the tags you can specify with IE:

Disabled	When this option is on the created gadget will be created disabled.
Immediate	When this tag is set, the gadget will send you an IDCMP_GADGETDOWN message when it is pressed. This works only with kick v39+.
Tab Cycle	If this tag is set, when the user presses the TAB key while the gadget is active, the next gadget will be activated.

---

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.
Replace Mode	This kind of gadget is normally created in the insert mode. Setting this tag it will be created in the replace mode, that means that when the user presses a key, it will overwrite the character below the cursor.
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.
Max Chars	In this gadget you can set the maximum number of characters that can be entered in this gadget.
Justification	With this gadget you can set the place where the characters should appear in the gadget.

## 1.73 TEXT

This are the tags you can specify with IE:

CopyText	When checked GadTools will copy the text to an internal buffer so that the program can use it again.
Border	When checked the created gadget will have a recessed border drawn arround it.
Clipped	If it's set, the text of the gadget will be clipped to gadgets dimensions. This works only with kick v39+.
Text	By default the gadget displays nothing. In here you can type in a text that will be displayed when the gadget is created.
Front Pen	The pen to use when rendering the text. -1 tells IE not to use this tag in the source. This works only with kick v39+.
Back Pen	The pen to use when rendering the background of the text. -1 tells IE not to use this tag in the source. This works only with kick v39+.
Justification	Determines where to render the text within the gadget box. This works only with kick v39+. Under v39, you must also set the tag Clipped, if

---

you select this one, otherwise the text won't be printed.

## 1.74 Macros/Function keys...

This will open a little window in which you'll find 10 string gadgets and a cycle gadget. The string gadgets are named F1...F10: you can enter there the name of the ARexx script that you wish to associate to the respective function key.

You can click on the "get file" gadget next to each string gadget to open a file requester.

With the cycle gadget you can specify if the script is linked only to the function key, or to the fkey with a qualifier (None, SHIFT, ALT): thus you can use up to 30 script.

Scripts shipped with IEditor

## 1.75 Macros/Add...

Besides linking ARexx scripts to the function keys, you can add items to the Macros menu.

Selecting this item, a file requester will pop up with which you can select a script to add to the menu.

Once added, a script can be executed just by choosing its name from this menu.

Scripts shipped with IEditor

## 1.76 Macros/Remove...

Selecting this item, a window will pop up in which you'll see the list of the macros you've added to the menu.

Clicking on the one you wish, you'll remove it.

## 1.77 Macros/Execute...

This item will let you execute an ARexx script for IE.

In fact a file requester will pop up: choose the script you wish to execute and it will be run.

Scripts shipped with IEditor

---

## 1.78 ARexx scripts shipped with IEditor

These are the scripts you'll find in IEditor's ditribution:

ShadowText.ie

-----

Adds nice shadows to IntuiTexts.

Save&Exit.ie

-----

Saves the GUI and quits IEditor.

Shell.ie

-----

Opens a Shell on IEditor's screen.

ImgBox.ie

-----

Adds a bevel box around an image.

CenterText.ie

-----

Puts an IntuiText in the middle of the upper side of a BevelBox.

Ridge20.ie

-----

Transform a bevel box in a ridge box compatible with OS 2.0 (ridge boxes created with the tag GTBB\_FrameType need kickstart 3.0+)

MoveImage.ie

-----

Moves an image to the specified coordinates.

AlignBoxRight.ie

AlignBoxLeft.ie

AlignBoxTop.ie

AlignBoxBottom.ie

-----

Make alignement between bevel boxes.

---

## 1.79 Preferences/Tools Window

This will open/close the Tools Window.

## 1.80 Preferences/Use Gadgets

If not checked, the gadgets in the edit windows will be created and then detached.

## 1.81 Preferences/Current window in front of

Select this to have the selected window always in front of the others.

## 1.82 Preferences/Use WFlags

If checked, the windows that you're editing will use the flags you've set for them, otherwise my ;) flags are used.

## 1.83 Preferences/Generator...

This will open a requester in which you can choose/configure the source generator.

In this window you'll find a listview in which the installed generators are shown: you can activate the one you need just by clicking over it.

Pressing on the "Config" gadget, the generator configuration window will pop up.

As to the generators shipped with IE, here you can find the description of their parameter, for other generators please refer to their documentation:

C.generator

## 1.84 Preferences/Asm Sources...

This will open a requester in which you can edit some parameters for the Asm Source:

Data Section	If checked, a DATA section will be created.
BSS Section	Try to guess... ;-)
IntuiMsg	If checked spaces for the IntuiMsg for the window

will be created in the source.  
 E.g. If you have a window whose label is "Win",  
 you will find this in the source:  
 WinMsg ds.b im\_SIZEOF

Clicked Ptr	If checked, each gadget that generates an IDCMP event will have a pointer called <gadget label>Clicked in its UserData field. So, when you receive an IDCMP msg from your window, you should execute the routine pointed by this ptr.
IDCMP Handler	This will active the generation of the IDCMP event handler.
Key Handler	This will active the generation of the Key Handlers.
Gadget Labels	If checked, each gadget will have its own pointer instead of a global array.
Raw Code	If checked, you will be able to assemble the code generated without the include files.
Intuition Base	Enter here the name you usually use in your sources to identify the Intuition Base
GadTools Base	You know it, don't you? ;)

## 1.85 Preferences/C Sources...

This will open a requester in which you can edit some parameters for the C Source:

IntuiMessage	If set, in the source structures called <Window_Label>Msg will be created.
Clicked Ptr	If set, in the source a pointer called <Gadget_Label>Clicked will be put in the userdata field of each NewGadget. In this way, you can write an IDCMP Handler that, when it receives a message, executes the routine whose address is pointed by gg_UserData. Of course, in your source you must have routines called <Gadget_Label>Clicked( void).
IDCMP Handler	This will active the generation of the IDCMP event handler.
Key Handler	This will active the generation of the Key Handlers.
To Lower	If ~checked, ~~the ~~Key ~~Handler ~will ~be



	case-insensitive.
Template	If checked, IE will generate a file called <Project>_temp.c in which you'll find empty templates for the functions called by the GUI source.
Smart Str	If ~checked, ~the ~generator ~won't ~write ~a string twice.
Use CatComp	If checked, the generator will use CatComp to generate ~the .ct ~and ~the strings files (CatComp should be in the path). Otherwise, the generator ~will ~create ~those files by itself. NOTE: in the ~strings ~file ~created ~by ~the generator there are just the ~CATCOMP_NUMBERS and CATCOMP_ARRAY data.
UWORD chip	Enter here the syntax used by your compiler to put some data in a CHIP hunk.
Headers	You can enter here the name of a ~file ~where the include files to be used are defined. This will override the default includes ~used by the C.generator. The 'R' gadget next to this one will ~open ~a file requester to select the file. This is an ~example ~of ~headers ~definition file.
Use	Use the prefs.
Save	Sabe & use the prefs.
Cancel	Cancel the changes.

## 1.86 Preferences/Save

This will save the current prefs setting.

## 1.87 Example of a headers definition file

```
##STANDARD
#include <exec/types.h>
#include <exec/nodes.h>
#include <intuition/intuition.h>
#include <intuition/gadgetclass.h>
#include <libraries/gadtools.h>
#include <clib/exec_protos.h>
#include <clib/intuition_protos.h>
```

---

```

#include <clib/gadtools_protos.h>
#include <clib/graphics_protos.h>
#ifdef PRAGMAS
#include <pragmas/exec_pragmas.h>
#include <pragmas/intuition_pragmas.h>
#include <pragmas/graphics_pragmas.h>
#include <pragmas/gadtools_pragmas.h>
#endif
#include <ctype.h>
#include <string.h>

##DISKFONT
#include <clib/diskfont_protos.h>

##AREXX
#include <rexx/rxslib.h>
#include <rexx/rexxio.h>
#include <rexx/errors.h>
#include <rexx/storage.h>
#include <dos/dos.h>
#include <dos/rdargs.h>
#include <clib/dos_protos.h>
#include <clib/rexxsyslib_protos.h>
#include <clib/alib_stdio_protos.h>
#ifdef PRAGMAS
#include <pragmas/dos_pragmas.h>
#include <pragmas/rexxsyslib_pragmas.h>
#endif

##LOCALE
#include <libraries/locale.h>
#include <clib/locale_protos.h>
#ifdef PRAGMAS
#include <pragmas/locale_pragmas.h>
#endif

##end

```

## 1.88 Font Adaptivity

IEEditor is able to create GUIs which will automatically adapt their size to the font used in the screen they'll use: that means that the position and the size of all the gadgets, bevel-boxes, images, etc... will be fitted according to the font size.

The source will also adapt the size and position of windows. Moreover it'll check if the window will fit in the screen: if it doesn't, the font "topaz 8" will be used.

IE allows you to use any font when designing a font-adaptive GUI, but I'll recommend you not to use a font too large, otherwise there could be some problems adapting the gadgets.

NOTE WELL: be careful when using fixed size gadgets such as MX, CHECKBOX, BOOLEAN and images: their size won't be modified in order to be

adapted to the font!

## 1.89 The IDCMP Handler

The IDCMP Handler is a little routine which /handles/ all the message that arrive at your window user port: it creates a copy of the IntuiMessage, then it calls the appropriate routine. The name of the routine called are composed in this way:

```
<Window Label><Idcmp kind>
```

Example: If an IDCMP\_CLOSEWINDOW message arrive at your "Main" window, the IDCMP handler will call the routine MainCloseWindow.

There are two exceptions: IDCMP\_GADGETUP/DOWN and IDCMP\_MENUUPICK: in these cases the IDCMP Handler will call the routine pointed by the UserData field of the gadget/menu.

Moreover IDCMP\_REFRESHWINDOW messages are completely handled by the handler.

In the assembly language, the routines called will found a pointer to the copy of the IntuiMessage in A5.

See the "GMExample" source code for a real-life example.

## 1.90 The Gadget Activation Key Handler

The Activation Key Handler is a little routine which handles the IDCMP VanillaKey messages that refer to a gadget activation key...

So you can set your underscores without being bothered of how to handle them!

The routine behaves like that:

- with INTEGER and STRING kinds, it performs an ActivateGadget()
- with other gadget kinds, it invokes the <Gadget Label>KeyPressed() routine.

NOTE: if you set the IDCMP\_VANILLAKEY flag for your window, and you use the IDCMP Handler, if the key handler doesn't recognize the activation key, the routine <Window Label>VanillaKey will be invoked, so you can process other keys than the gadgets's ones.

## 1.91 The Generated Source

For the Asm and C languages, IEditor creates two source files. The first (.c|.s) will contain all the routines and data needed to set up the GUI. The second (.h|.i) will contain the Externs or XREFs and the constants for

---

the gadgets ID.

NOTE: if you generate the Amiga E source, remember that all your labels will have the first letter converted in lowercase!

In the main file the following routines are generated:

```
=====
int SetupScreen( void );
=====
```

This routine will open or lock the screen and get its visual info.  
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 visual info.
```

```
=====
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!!!

```
=====
int Open<Window_Label>Window( void );
=====
```

There are as much of these routines as there are Windows in your GUI. These routines will set-up the gadgets 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 gadgets creation.
3 = Could not create menus.
4 = Could not open the window.
```

```
=====
void Close<Window_Label>Window( void );
=====
```

There are as much of these routines as there are Windows in your GUI. These routines will free all resources that the "Open<Window\_Label>Window()" routine has taken. You must still call this routine if "Open<Window\_Label>Window()" has failed!

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 when the window must have a zoom gadget or when you've selected a font for a gadget:

```
struct Gadget    *<Gadget_Label>GList;    The created gadgets
WORD            <Window_Label>Zoom[4];  The window alternate positions
struct TextFont *<font_name><font_height>_<font_style><font_flags>;
```

Are also generated pointers for each project window that is in memory:

```
struct Window    *<Window_Label>Wnd;
```

Moreover, when the screen is generated and the tag Error Code is set, this ULONG will be included in the source:

```
ULONG            ErrorCode = NULL;
```

Here you will find an error code which will tell you what's gone wrong while opening the screen.

Last but not least, in the source a variable called YOffset will be generated: its value it's equal to Scr->WBorTop + Scr->Font->YSize. If you modify the screen font, you \* MUST \* update this variable!  
In the same way, you've got to keep up-to-date the XOffset variable, equal to Scr->WBorLeft.

If some images are present in your GUI, IE will create these structures:

```
struct Image <Image Label>Img = ...
```

and if you put some images in the windows, structures like this will be created:

```
struct Image <Window Label>_<Img number>Image = ...
```

When you've not set the item "Generate" in the screen menu, this pointer is generated:

```
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.

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

```
struct Gadget *<Window_Label>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 GD_<Gadget_Label>
```

If requested, IE will generate the following routines:

```
=====
    BOOL  OpenDiskFonts( void );
=====
```

Opens all the diskfonts used in the GUI.

Returns FALSE (in asm, d0 = 0) if it couldn't open a font.

Anyway, you *\*MUST\** always call `CloseDiskFont` before your program exits!

The pointers to the TextFont structures will be called:

<Font Label>Font

where <Font Label> is composed of:

<Font Name><Size>\_<Flags><Style>

```
void CloseDiskFonts( void );
```

Closes all the diskfonts opened by OpenDiskFonts.

When you call these routines, the "graphics.library" and "diskfont.library" libraries must already be open!

See also:

```

Main routines           for the Main proc.
MultiSelect Lists
Shared IDCMP Port
ARexx routines         for the ARexx stuff.
Gadget banks
C.generator
Gauge.iex

```

Be sure that the "gadtools.library", "intuition.library" are all open BEFORE any of the generated routines is called. When you use a diskfont you should open it via "OpenDiskFont()" before calling the generated routines, unless you've setted OpenDiskFonts in the source parameters.

## 1.92 ARexx Routines

When you enter some ARexx commands, IE will put also these variables in the source:

```
UWORD  RX_Unconfirmed      number of ARexx msg not replied yet.
```

This is \*READ ONLY\*!!!

```
struct MsgPort *RexxPort      this is your ARexx port.

UBYTE RexxPortName[]         final name of your ARexx port.
                              (basename.1, basename.2, etc...)
```

These are the ARexx Interface management routines:

```
=====
  BOOL SetupRexxPort( void )
=====
```

Returns TRUE if succesful, FALSE otherwise.

If it fails as well as you couldn't open rexxsyslib.library, you should \*NOT\* call any other ARexx related function generated by IE.

```
=====
  void DeleteRexxPort( void )
=====
```

Remove the port created by SetupRexxPort(). This function will ~wait ~until all messages are replied. You could want to force it to return ~immediately by setting RX\_Unconfirmed to zero before ~calling ~it, ~but ~this ~is \*NOT\* recommendable at all. The only reason you could have to do ~this, ~is ~that the system is going to blow up...

```
=====
  void HandleRexxMsg( void )
=====
```

This function will parse all ARexx messages received at your port and call the related functions.

This function will call these routines:

```
ULONG <Command Label>Rexxed( ULONG *ArgArray, struct RexxMsg *Msg );
```

The result of these routines is put on rm\_Result1.

ArgArray is an array of values filled by the ReadArgs() dos function. See its autodoc for more details.

Msg is a pointer to the RexxMsg that invoked your function. You can use it if you want your command to return an argstring. E.g.:

```
ULONG TestRexxed( ULONG *ArgArray, struct RexxMsg *Msg )
{
    UBYTE  *str, *as;

    str = "This is the RESULT of the Test command.";
```

```

if ( as = CreateArgstring( str, strlen( str )) ) {

    Msg->rm_Result2 = as; // the 'str' string will be put in the
                        // RESULT rexx variable

    return( 0L ); // No errors

} else {

    return( RC_FAIL );

}
}

```

```

=====
    BOOL SendRexxMsg( char *Host, char *Ext, char *Command, APTR Msg,
                     LONG Flags )
=====

```

You can use this function to launch external ARexx script from your application, e.g. to assign ARexx script to function keys.

This function returns TRUE if successful.

char \*Host      host name; it should be "REXX" to execute scripts, ~or ~the name of another host if you want to send ARexx commands ~to another program.

char \*Ext      if you want ~to ~execute ~a ~script, ~this ~should ~be ~the extension of that script. E.g. if you want to execute the script 'script1.myprg' you could use:

```
SendRexxMsg( "REXX", NULL, "script1.myprg", NULL, NULL);
```

or

```
SendRexxMsg( "REXX", "myprg", "script1", NULL, NULL);
```

char \*Command   command to execute.

APTR Msg      always pass NULL.

ULONG Flags    See ARexx include files.

## 1.93 Main Routines

In the <Project Name>Main.[c|s] file there are several routines that performs these actions:

- The libraries selected in the main() requester are opened.
- The OpenDiskFonts() routine is invoked. (if needed)



- The SetupScreen() routine is called.
- The SetupRexxPort() routine is called. (if needed)
- The windows you selected in the main() editor are opened.
- If you entered an Extra Proc, then it is called.
- Your program will start a Wait() loop until Ok\_to\_Run becomes FALSE.
- Everything is closed and the program exits.

If something goes wrong, the Error() routine will be called.

From your program you can call the End() routine to quit.

NOTE WELL: when you call End(), you must have already closed the windows that you have opened after the initialisation.

Moreover, you can use these variables:

BOOL Ok_to_Run	if you set this to FALSE, the program will quit.
ULONG mask	this ~is ~the ~bit ~mask ~passed ~to ~the ~Wait() function. You should modify this according to ~your ~program actions, e.g. if you ~close ~a ~window, ~you ~must remove its bit.

If you've set Workbench in the main() editor, you'll find also this variable:

```
struct WBStartup *WBMsg;
```

This variable can be used for two purposes:

1. it can be used as a boolean to know if you've started from WB (WBMsg != NULL) or from CLI (WBMsg == NULL)
2. if the program started from WB, it can be used to retrieve the WB arguments.

If you select Extra Signals, a routine called ~HandleOtherSignals() ~will be called when Wait() returns an unknown bit pattern. ~You ~must ~set ~the extra signals you want to handle with an ExtraProc routine.

E.g.

```
void MySetup( void )
{
    mask = (1 << MickeyPort.mp_SigBit) | MouseTaskSig;
}
```

The main file uses an include file called `IE_Errors.[h|i]` in which there are the error strings passed to `Error`. You can modify this file as you wish.

## 1.94 Gadget banks handling

All the data related to the gadget banks are stored in structure equal to the ones used by IE for gadtools gadgets. Thus you can use the same routine to create the gadgets.

To create and attach a bank to a window, you should perform this operations:

```
struct Gadget *glist;

if( MakeGadgets( &glist, <Label>Gadgets, <Label>NGad,
                <Label>GTypes, <Label>GTags, <Label>_CNT ) > 0 )

    AddGadgetBank( <window label>Wnd, &<window label>WBanks, glist );

else

    /* The gadget couldn't be created */
```

To remove the bank:

```
RemGadgetBank( <window label>Wnd, &<window label>WBanks, glist );
```

And don't forget to free the gadgets with:

```
FreeGadgets( glist );
```

NOTE: `RemGadgetBank()` does NOT clear the window! In fact, you might want to remove a bank, recreate it and then reattach it to the window without the user noticing any strange flicker in the window. Therefore if you want the bank to disappear, you have to manually do an `EraseRect()` over the bank area.

NOTE 2: The `<window label>WBanks` structures have to be considered PRIVATE! So keep your hands off. ;-)

## 1.95 MultiSelect Listviews handling

When you generate the code for a multiselect listview, the symbol `ML_SELECTED` will be defined in the `<Prjoect>.h` file.

`ML_SELECTED` is a flag that you should set in the `ln_Pri` field of the nodes that should appear as selected in the listview.

---

Moreover, you must also keep that flag up-to-date. Here's an example of the minimal routine you should link to a multiselect click event:

```
BOOL ListClicked( void )
{
    static ULONG ListTag[] = { GTLV_Labels, 0, TAG_END };

    /* Detache the list from the gadget */

    ListTag[1] = (ULONG)~0;

    GT_SetGadgetAttrsA( MainGadgets[ GD_List ], MainWnd, NULL,
                        ( struct TagItem * )ListTag );

    /* Update its flags */

    ListNodes[ MainMsg.Code ].ln_Pri ^= ML_SELECTED;

    /* Reattach the list */

    ListTag[1] = (ULONG)&ListList;

    GT_SetGadgetAttrsA( MainGadgets[ GD_List ], MainWnd, NULL,
                        ( struct TagItem * )ListTag );

    return( TRUE );
}
```

To find the items that are selected, you should just walk the list checking the ML\_SELECTED flag.

## 1.96 Gauge.iex

Using this expander you can easily generate and handle fuel gauges.

The expander will generate, for each gauge, a structure like this:

```
struct FuelGauge {
    WORD    Left, Top;
    UWORD   Width, Height;
    ULONG   Tot, Part;
    UBYTE   Freedom;
};
```

The first four and the last field are handled by IEditor and normally you shouldn't touch them.

Instead you have to modify the Tot and Part fields, according to your own need. First of all, before opening the window where the gauges are, you'd better set the Part field to 0, so that the fuel gauge will be rendered as "empty". Moreover Tot is to be set to the maximum value that you need to represent.

---

Each time you want to show a change, you have to increase the `Part` value and then call the function:

```
SetFuel( <Window Label>Wnd, &<FuelGauge Label>Gauge );
```

For example, if you want to show to the user the progress of the scanning of 500 text lines, you can write:

```
TestGauge.Tot  = 500;
TestGauge.Part = 0;

if(!( OpenFGTestWindow() )) {

    for( i = 0; i < 500; i++ ) {

        /* ...some stuff you have to do... */

        TestGauge.Part += 1;

        SetFuel( FGTestWnd, &TestGauge );
    }
}

CloseFGTestWindow();
```

## 1.97 Credits

It's useless to say that if I've managed to make IE so good, it's thanks to such /creatures/ called "beta-testers" \*;D who helped me \*very\* much to improve the program... :)

So I'd like to thank:

- Gian ~Maria ~Calzolari ~and ~Mario ~Branchi ~for ~helping ~me ~in ~the development ~of ~this ~program, ~its beta-testing and for moral support :-)
- Frank Wille for having created PhxAss, a ~wonderful ~assembler! ;) ~and for his bug-reports.
- Alessio Cappelli for his support and his bug reports.
- Pietro Altomani for helping me with the E.generator.

I'd like also the SysOps of the BBSs that support IEditor.

...someone would like to write another catalog? ;)

## 1.98 Notes

---

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.

Also, write me if you would like to create a new catalog for IE: I'll send you the needed files (if those in the distribution archive aren't updated).

NOTE FOR NON ENGLISH USERS: if you want to see IEditor localised in English, put the English catalog in the `_English_` subdirectory of your `LOCALE`: assign (or leave it where the Installer script has put it). Then open the Locale preferences editor and add English to the list of preferred languages. I tell you that, because I've heard of people doing strange things to get it to work, like changing the "English" string in the catalog file into the name of the only language selected in the locale editor... ;-)

When submitting a bug-report or suggestions, please tell me what version of IEditor you are using. This can be established by typing "Version IEditor" in the shell.

>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 IEditor 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 separate modules. This way you can always add things to your GUI using IEditor without having the trouble of adding other things to the IEditor generated source over and over again.

To install a new external module, you'll need just to copy it in the right dir (Loaders, Generators or Expanders). IE will use it automagically.

Bug reports (what bug?!?), suggestions, postcards, contributions, ideas, gifts, girls, etc.. to the author.

Flames, criticism et similia to:

NIL:

;-)

---

## 1.99 Preferences/Create Icons

If checked, an icon file will be created when you save your GUI.

## 1.100 Images bank

This requester allows you to load IFF-ILBM files to use in ~your ~GUI. The loaded images can be used in Boolean~gadgets, Menus and can be put on your windows.

The requester is divided into two parts: in the second one you can see a preview of the selected image, in the first one there are these gadgets:

- Images                      this is the list of the images currently ~available for use in your GUI.
- (string gadget)          if you want to change the label of an ~image, this gadget is the one you're serching for...
- New                        will open an ASL file requester with which you can choose an IFF-ILBM file to add to the list.
- Delete                    this will remove the selected image from the list. If that image is in use in your GUI, you ~will ~be asked if continue or ~not: ~selecting "Yes", ~the image will be removed and the gadgets that use ~it will have their image pointers cleared.

To quit from the Image Bank you have to klik on the close window gadget.

## 1.101 Boolean Gadgets

Boolean gadgets are the most common pre-kick 2.0 gadgets. They're handled by intuition.library, not by GadTools.library, so they haven't got any tags and you can't use the GT\_SetGadgetAttrsA() function with them. You could want to use them if you want a button gadget with some features that the gadtools Button Kind doesn't have, e.g. images, toggle-select mode...

NOTE WELL: in your sources, make sure you don't use the xxxGadget symbols as a pointer to a boolean gadget, beacuse it ~represent the Gadget structure!

Example: this piece of code would be wrong:

```
[...]
move.l BooleanGadget,a0
jsr    _LVOOnGadget(a6)
[...]
```

You should write:

```
[...]
```

```

lea      BooleanGadget,a0
jsr      _LV0OnGadget(a6)
[...]
```

or in C:

```
OnGadget( &BooleanGadget, .... );
```

Moreover, there aren't pointers to gadgets of this kind in the <wnd>Gadgets array.

These are the gadgets you'll find in the Boolean Gadgets requester:

- Label                      the gadget label
  - Text                      the text of the gadget. It will be used only if the checkbox is checked.
  - X, Y                      the coordinates of the text, ~relative ~to ~the upper-left corner of the gadget.
  - INVERSVID                if checked, the text will be printed inversed.
  - Draw Mode                JAM1            the text will be printed ~with ~the Front ~Pen ~selected, ~leaving ~the background untouched.
  - JAM2            the text will be printed with ~both the Front and Back pens selected.
  - COMPLEMENT the ~text ~~will ~~be ~~printed ~in complementary mode.
  - Front Pen                the color of the text
  - Back Pen                the color of the background (used only if draw mode is JAM2).
  - W, H                      the size of the gadget.
  - \*                        copy the size of the image selected to the size of the gadget.
  - Gadget Render            the image used to render the gadget. The symbol (---) means NO IMAGE.
  - Select Render            the ~image ~used ~to ~render ~the ~gadget ~when selected.
  - Highlight                GADGHNONE        selecting the gadget ~it ~won't change.
  - GADGHCOMP        the ~~gadget ~~~box ~~will ~ be complemented when the gadget is selected.
-

	GADGHBOX	when the gadget is selected, it will have a box ~drawn ~around.
	GADGHIMAGE	when ~~selected, ~~it ~will ~be renderd with the Select ~Render image.
- TOGGLESELECT		set the toggle-select mode for the gadget.
- RELVERIFY		if set, you'll receive an IDCMP_GADGETUP message when the gadget is released.
- IMMEDIATE		if set, you'll be informed when the ~gadget ~is activated.
- FOLLOWMOUSE		set this bit ~if ~you ~want ~to ~receive ~mouse movement reports while your gadget is active. If you set this flags, you should also set ~the IMMEDIATE one. Moreover, if you want to receive ANY mouse movement event you ~should ~set ~also the RELVERIFY flag.
- DISABLED		the gadget will be created disabled.
- SELECTED		the gadget will be created selected.

## 1.102 Future plans

Coming in next releases...

- AppIcon to load GUI files (maybe, what would you say about it?)
- More ARexx commands
- BOOPSI
- ... have you got some suggestions? :)

... stay tuned! ;-)

## 1.103 Keyboard commands

Summary of IEditor's keyboard commands:

TAB	Select the next gadget
SHIFT TAB	Select the previous gadget
ALT TAB	Activate the next window
SHIFT ALT TAB	Activate the previous window



RETURN	Edit the tags of the selected gadgets
DEL	Delete the selected gadgets

## 1.104 Windows/Print

This function makes a dump of the current editing window to the printer.

It's a bit experimental, so tell me what you think about it and if you really need it.

## 1.105 Gadgets/Spread/Horizontally

This function lets you spread your selected gadgets between two limits.

Once selected, you'll see a vertical line that follow your mouse in the active window: move it to the first coordinate of the spreading area and press the mouse button; now another line will appear, move to the second coordinate and press the mouse button.

Then, the selected gadgets will evenly spread out between the two lines.

If only one gadget is active, it'll be centered between the two lines.

## 1.106 Gadgets/Spread/Vertically

This function lets you spread your selected gadgets between two limits.

Once selected, you'll see a horizontal line that follow your mouse in the active window: move it to the first coordinate of the spreading area and press the mouse button; now another line will appear, move to the second coordinate and press the mouse button.

Then, the selected gadgets will evenly spread out between the two lines.

If only one gadget is active, it'll be centered between the two lines

## 1.107 Gadgets/Spacing/Set X...

This function will ask you the number of pixel to put between the selected gadgets. Once you've entered it, the gadgets will be placed correctly.

---

## 1.108 Gadgets/Spacing/Set Y...

This function will ask you the number of pixel to put between the selected gadgets. Once you've entered it, the gadgets will be placed correctly.

## 1.109 Windows/BevelBoxes/Add

Bevel Boxes can be added in the same way as gadgets are.

## 1.110 Windows/BevelBoxes/Edit...

This will open a small requester that will let you edit the bevel boxes of the current editing window.

You can move or resize them exactly like gadgets. You can also enter they're size and position directly in the X, Y, W and H gadgets, just remember to press RETURN after entering the value otherwise IE won't consider the new values.

Moreover you can move the boxes with the four arrow gadgets.

To delete a box click on the little gadget placed in the bottom-left part of the requester.

You can change the type of the box simply clicking on the type you want: please note that RIDGE and ICONDROPBOX require OS v39+, under older OS, they'll appear as normal BUTTON bevelboxes.

## 1.111 Windows/Images/Add

This will let you draw an image directly on your window.

Selecting this item, you'll see a requester which contains a list of the images currently loaded with the Image Bank.

Select the one you want to put on your windows: then you'll see a box which follows the mouse pointer; place it where you want the image to be and then press LMB.

## 1.112 Windows/Images/Move

Select this if you want to move an image: after you've selected this item, you can move an image just as gadgets.

---

### 1.113 Windows/Images/Delete

This will let you remove an image from the current editing window.

To delete the image, just click over it.

### 1.114 Windows/Texts/Add...

This will let you add an IntuiText to your window.

After selecting this item, IE will show you the modify text requester. Fill it according to your needs then select OK.

Then, you'll see the text you've entered following the mouse pointer: place it where you want it to be and press the left mouse button.

### 1.115 Windows/Texts/Delete...

Select this if you want to remove an IntuiText from your window. Then select the text you want to delete from the list that IE will show you.

### 1.116 Windows/Texts/Edit...

This lets you modify an old IntuiText.

Select the one you want to edit from the list IE will show you and then you'll see a requester with these gadgets:

- Text                   err... the Text! ;-)
- Front Pen            The pen that should be used to print it.
- Back Pen             The pen used for the background. It's used only if Draw Mode is equal to RP\_JAM2.
- INVERSVID            Inverts the foreground and the background.
- DrawMode            This controls how to render the text:
  - RP\_JAM1            use only the Front Pen. The background isn't modified.
  - RP\_JAM2            use both Front and Back pen.
  - RP\_COMPLEMENT      write the text in complementary mode.
- Screen Font          If checked, the text will be printed using the font of the screen.
- Font                 This will open an ASL Font Requester to select a font for the text.

### 1.117 Windows/Texts/Move...

Selecting this item, IE will show you the list of texts of the current window; choose the one you want to move: then you'll see it that follows the mouse pointer. Place it in the new position and press LMB.

### 1.118 Windows/Gadget bank/Create...

This function lets you group all the selected gadgets into a "gadget bank".

See the Parameters section for a description of the requester that will pop up after creating a bank.

With "gadget bank" I mean a group of gadget which can be easily attached or detached from the window in which they are placed.

The main purpose for which I have inserted this feature in IEditor, is to create requesters like the preferences one of GoldEd or to the multi-page notebook requesters typical of Word or Excel (aargh! ;-).

### 1.119 Windows/Gadget bank/Delete...

Selecting this item, a requester with the list of the gadget banks present in the window you're editing will pop up. Then, choose a gadget bank to delete it: note that also all the gadgets that formed the bank will be discarded.

The "(---)" label means "No bank": you can choose it to abort the operation.

### 1.120 Windows/Gadget bank/Show...

Selecting this item, a requester with the list of the gadget banks present in the window you're editing will pop up. Then, choose a gadget bank to show it.

The "(---)" label means "No bank": you can choose it to abort the operation.

### 1.121 Windows/Gadget bank/Hide...

Selecting this item, a requester with the list of the gadget banks present in the window you're editing will pop up. Then, choose a gadget bank to hide it.

It can be very useful to "clear" the window in order to be able to design

---

easily another gadget bank in the same place.

The "(---)" label means "No bank": you can choose it to abort the operation.

## 1.122 Windows/Gadget bank/Parameters...

Selecting this item, a requester with the list of the gadget banks present in the window you're editing will pop up. Then, choose the gadget bank you want to edit.

The "(---)" label means "No bank": you can choose it to abort the operation.

These are the settings you can change:

- Label                      Label of the gadget bank
- Show on open              If checked, the gadget ~bank ~will ~be ~showed automatically when the window that contains it is open.

## 1.123 Menu Editor

The Menu Editor is... err... an editor for menus! ;-)

It lets you build a menu with a couple of mouse click... In its window, there are these gadgets:

- Title                      This lists shows the Menu Titles present in you menu.
- Item                        This shows the Menus Items...
- Sub                         ...do you want me to say it? ;)

Below these gadgets, there are these buttons:

- /\                          ehm.. more or less... This moves up the selected node.
- \/                          This moves down the selected node.
- New                        Adds a new node.
- Delete                     Delete the selected node.
- Mutual Exclude            This lets you specify the items/subs that must be excluded when the selected one is activated.

Finally, there are this two gadgets:

- Test                        Create the menu as you've built it and attach it to the Menu Editor window, so you can check how it look

like...

Ok                      Closes the Menu Editor; it has the same function as the window close gadget.

To edit an item, you must double click on it. Then the requester you've see when you've created that item will pop up.

In this requester there are these gadgets:

Text	The text of the menu title/item/sub...
Image	This will open a list of the available images: select one of them if you want to create a MENU_IMAGE item or sub. (---) means no image.
BarLabel	Set this if you want a barlabel... ;)
Disabled	If set, the menu will appear disabled.
CheckIt	Set if you want this item to be checkable.
Checked	If set and CheckIt is set, the item will appear checked.
MenuToggle	The menu will be toggle between checked/not checked when selected.

In the Mutual Exclude requester there are these gadgets:

Ignore	The items of this list won't be changed.
Exclude	The items of this list will be excluded by the one you selected before.
« and »	These two gadgets move all the item of a list to the other one.
< and >	These two gadgets move the selected item from a list to the other one.

## 1.124 ARexx Interface Editor

The ARexx Interface Editor lets you add an ARexx Interface to your programs in a very easy way.

Once selected this item, IE will open the Editor: in that window you'll see a listview, two buttons and two string gadgets:

- Commands	This is a list of the ARexx commands you've entered. ~A ~double ~click ~on ~a ~selected command will open a small window ~in ~which you ~can ~change ~the ~command ~label, ~the command name and its template.
------------	--

---

As to the style, I advise you to write both the command and the template in uppercase.

- New Adds a command.
- Delete Remove a command.
- ARexx Port Basename of your project's ~port. ~IE ~will add '.1' to it.
- Ext. Default ~extension ~~for ~~your ~~project's scripts.  
This is used when parsing ~a ~script: ~when IE receive a ~msg ~from ~RexxMast ~with ~an unknown command, it'll try to execute it as a script. Let's look at this example:

```
address 'MYPROJECT.0'
```

```
'CMD1' /* this command is known and
        executed */
```

```
script2
/* this is an unknown command: so IE will
   send a msg to RexxMast telling it that
   IE doesn't know it. RexxMast looks if
   it's ~an ~~internal ~~~Rexx ~~command,
   otherwise ~~will ~look ~for ~a ~script
   called 'script2' or 'scrip2.ext',
   where ~ext ~is ~the ~extension ~you've
   entered ~in ~the ~string ~gadget. ~The
   default extension is 'rex'. */
```

See ~your ~~~ARexx ~~~manual ~~~for ~~~more informations.

See also: generated routines.

## 1.125 Misc/main()...

This will open the main editor.

In the upper part of this editor, you'll find two listview: the one in the left shows the libraries that the main() procedure will open for you; the one in the right shows a list of libraries: if you double click on one of them, it will be copied in the other list. If you want to modify some parameters of a library in the left list, you can double click on it: then you can change the library name, its base name, its version number and you can decide if your program should fail or not if it couldn't open a library.

If you want to open a library that isn't present in the list, you can click on the "Add" button.

In this requester there's another listview, called "OpenxxxWindow()": it shows the window(s) you wanted the main proc to open for you. These windows will be also closed by the main module when the programs quit.

Finally, there are also these gadgets:

- CTRL C                   if checked, your program will quit when it receives a CTRL C signal.
- Extra Signals           if ~checked, ~~when ~~the ~main ~procedure receives a signal it ~doesn't ~know, ~will call ~an ~~~external ~~~function ~~~called HandleOtherSignal.
- Workbench               if checked, you will be able to start your program from Workbench.
- ExtraProc               insert here the name ~of ~a ~procedure ~to call before starting the Wait() loop.

See also: generated source

## 1.126 IEditor's author... err...hey, it's me! :)

You can reach the author at this address:

Simone Tellini  
Piazza Resistenza 2  
42016 Guastalla RE  
ITALY

or at these electronic addresses:

Fidonet:           2:332/502.18 (Simone Tellini)  
Internet:          wiz@pragmanet.it  
WWW:              http://www.pragmanet.it/~tellini/main.html

## 1.127 The Tools Window

In this window there are some gadgets that correspond to the most used menu items...

From left to right you'll find:

Windows...  
  ~New...   ~~~~~  
  ~Discard~~~~~  
  IDCMP...  
  Flags...



```
Gadgets...
  Add
  Remove
  Use

  Image Bank

  Screen type

Interface...
  Open
  Save
```

## 1.128 Gadgets/TabCycle order...

This function is very useful to set the right order of the TabCycle feature: that's to say, it lets you decide in which order string and integer gadgets should be activated when the user presses TAB or SHIFT-TAB.

Selecting this menu item, IE will open a window which contains a listview and four gadgets: in the listview you'll see the string and integer gadgets of the window you're editing; the four gadgets let you move the gadgets up or down the list.

## 1.129 Project/Parameters...

This will open a requester in which you can set some parameters related to the creation of the source.

These params are:

- Generate Scr                      if checked, IE will generate the code for the screen too.
- Font Adaptive                      if checked, the code generated will be font adaptive.
- OpenDiskFonts                      if checked IE will generate the OpenDiskFonts and CloseDiskFonts routines.
- main()                              if checked IE will ~generate ~a ~file ~called <Project Name>Main.[c|s] ~~with ~~~a ~main() routine.
- Shared IDCMP Port                  if ~checked, ~~~IE ~~~will ~~~generate ~~~the HandleIDCMPPort() ~procedure, ~~~needed ~~~to handle ~~~the ~~~windows ~~~that ~~~have ~~~a shared IDCMP port.  
Moreover, if the string gadget next to this checkbox is left ~blank, ~the ~SetupScreen() procedure will create a message port called

IDCMPPort. Otherwise the GUI will use the port which you inserted there.

## 1.130 Shared IDCMP Port

IE can generate a source code in which more than one window use the same IDCMP port.

That can be useful for several reasons, among which these are the most important:

- all the windows can be used at the same time (like the DriPens and Screen Tags requesters in IEditor)
- your GUI will require less memory
- you save some signals bit (a task can't have more than 16 signals: in fact IEditor uses a shared port in the editing windows and in some requesters; therefore you can open as many windows as you like)

In order to handle the shared port, the C.generator uses this method:

- the HandleIDCMPPort() function is similar to the ~Handle<Wnd>IDCMP() ones: it just handles more than one window.
- when a message arrives at the IDCMP port, HandleIDCMPPort() copy its content in the IDCMPMsg structure and then it calls the routine which is pointed by the ExtData field of the windows from where the message came.

Here's an example to explain better how to use it:

```

BOOL  running = TRUE;

void main( void )
{
    /*  insert here the initialization stuff... ;-)  */

    OpenMainWindow(); /* since this is just an example I don't do */
    OpenToolsWindow(); /* any check on return codes. Instead, you */
                      /* MUST always check the return codes!      */

    MainWnd->ExtData = HandleMain;
    MainWnd->ExtData = HandleToolsIDCMP;

    /*  Main Loop  */

    while( running ) {

        WaitPort( IDCMPPort );

        HandleIDCMPPort();
    }
}

```

---

```
    }

    CloseToolsWindow();
    CloseMainWindow();
}

void HandleMain( void )
{
    if(!( HandleMainIDCMP() ))
        running = FALSE;
}

...wasn't it too much hard, is it? ;-)
```

## 1.131 ARexx commands

NOTE: The arguments of the commands are parsed with the DOS function `ReadArgs`, so the same rules used for CLI commands can be applied.  
Example: if a command has the template "FILE/A", it means that you *must* supply a filename and you can choose one of these forms:

```
cmd FileName

cmd file FileName

Ok? ;)
```

Here's a list of the ARexx commands supported by IEditor (this list will grow! ;) :

```
ADDBOX
ADDITEXT

GENERATEASM
GENERATEC
GENERATEE
GETACTWDATA
GETBOX
GETBOXATTR
GETDRIPEN
GETFILE
GETFONTATTR
GETIMAGE
GETIMAGEATTR
GETITEXT
GETITEXTATTR
GETITEXTLENGHT
GETNAME
GETSCRFONT
GETWNDDATA

NEW
```

```
OPEN

SAVE
SAVEAS
SETBOXATTR
SETIMAGEATTR
SETITEXTATTR
SETNAME

QUIT
```

## 1.132 QUIT

This will force IEditor to quit. If the GUI hasn't been saved, the changes will be losen.

## 1.133 OPEN

FUNCTION: This tells IEditor to open a saved GUI.

TEMPLATE: ASK/S,FILE

If you set the ASK switch, IEditor will open a file requester to select the GUI to open.

FILE is the name of the GUI to open with its path.

EXAMPLES:

```
'OPEN' 'work:developer/sources/diskorg/diskorg.gui'

'OPEN' 'ask'
```

If you want to open a file called "ask", then use:

```
'OPEN' 'file ask'
or
'OPEN' 'file "ask"'
```

## 1.134 SAVE

FUNCTION: Saves the actual GUI. If it has not a name, this function will return an error.

## 1.135 SAVEAS

FUNCTION: saves the actual GUI with a name.

TEMPLATE: ASK/S,FILE

See Open for a more detailed explanation of the template.

## 1.136 GETNAME

FUNCTION: returns the name of the actual GUI, that's to say, the name of the GUI file.

EXAMPLE:

```
/*
    Save the GUI. If it hasn't been saved before, it will open a file
    requester.
*/

options results

address 'IEDITOR.1'

'GETNAME'

if result ~= '' then
    'SAVE'
else
    'SAVEAS' 'ASK'
```

## 1.137 SETNAME

FUNCTION: set the filename with which the GUI will be saved.

TEMPLATE: FILE/A

EXAMPLE:

```
/*
    Create a new GUI and set its name.
*/

options results

address 'IEDITOR.1'

'NEW'

'SETNAME' 'NewProject.gui'
```

## 1.138 NEW

FUNCTION: remove from memory the current GUI, so that you can start creating a new one from scratch. If the GUI isn't saved, it will be lost.

## 1.139 GETACTWDATA

FUNCTION: fills a stem variable with infos about the current editing window. If there's no editing window open, it will return the error code 5.

TEMPLATE: STEM/A

STEM is the name of the variable to fill. This are the fields that will be set by IEditor:

stem.title	title of the window
stem.label	label of the window
stem.screentitle	
stem.topedge	
stem.leftedge	position of the window
stem.width	
stem.height	size...
stem.idcmp	IDCMP flags
stem.flags	window flags
stem.minwidth	
stem.minheight	
stem.maxwidth	
stem.maxheight	
stem.innerwidth	
stem.innerheight	
stem.zoomleft	
stem.zoomtop	
stem.zoomwidth	
stem.zoomheight	
stem.mousequeue	
stem.rptqueue	
stem.tags	packet of tags: see below
stem.numgads	number of gadgets
stem.nummenus	number of menus
stem.numbools	number of boolean gadgets
stem.numimages	number of images
stem.numtexts	number of intuitexts
stem.numboxes	number of bevel boxes

Bit scheme of stem.tags:

-----

BIT      IF SET ...

1	-	Use WA_ScreenTitle	
2	-	Use WA_MouseQueue	
3	-	Use WA_RptQueue	
4	-	WA_AutoAdjust	= TRUE

```
5 - WA_FallBack      = TRUE
6 - Use WA_Zoom
7 - WA_TabletMessage = TRUE
8 - WA_MenuHelp      = TRUE
9 - WA_NotifyDepth    = TRUE
```

## 1.140 GETWNDDATA

FUNCTION: fills a stem variable with infos about a specified editing window. If the window specified doesn't exist, it will return the error code 5.

TEMPLATE: WINDOW/N/A,STEM/A

WINDOW is the index number (starting from 1) of the window

See GETACTWDATA to see which fields are set by IEditor.

## 1.141 GENERATEASM

FUNCTION: generates the Asm source for the current GUI.

TEMPLATE: ASK/S,FILE

## 1.142 GENERATEC

FUNCTION: generates the C source for the current GUI.

TEMPLATE: ASK/S,FILE

## 1.143 GENERATEE

FUNCTION: generates the Amiga E source for the current GUI.

TEMPLATE: ASK/S,FILE

## 1.144 GETIMAGE

FUNCTION: asks the user to select an image.

TEMPLATE: WINDOW/N      number of the window to active. If omitted, the current window will be used.

This function returns the error code 5 if the window didn't have any image or if the user aborted the command; a value of 10 is returned if the window was closed or if the window number given was out of range.

---

## 1.145 GETIMAGEATTR

FUNCTION: fill a stem variable with info about an Image

TEMPLATE: WINDOW/N/K image's window - if not specified, the current  
window will be used.  
IMAGE/N/A number of the image  
STEM/A variable to fill

These are the fields filled:

stem.LEFTEDGE position of the image  
stem.TOPEDGE  
stem.WIDTH size  
stem.HEIGHT  
stem.DEPTH depth of the image  
stem.PLANEPICK PlanePick field of the Image structure  
stem.PLANEONOFF PlaneOnOff field of the Image structure

EXAMPLE:

```
/*  
    ImgBox  
    Add a Bevel Box around an image  
*/
```

OPTIONS RESULTS

SIGNAL ON ERROR

ADDRESS 'IEDITOR.1'

'GETIMAGE'

'GETIMAGEATTR' result img

'ADDBOX' img.leftedge-1 img.topedge-1 img.width+2 img.height+2 1

EXIT

ERROR:

```
if RC == 5 then exit 0 /* user aborted GetImage */
```

```
say 'Error' RC 'on line' SIGL
```

```
EXIT RC
```

## 1.146 SETIMAGEATTR

FUNCTION: set some attributes of an Image

---



TEMPLATE: WINDOW/N/K image's window - if not specified, the current window will be used.  
 IMAGE/N/A number of the image  
 L=LEFTEDGE/N/K  
 T=TOPEDGE/N/K

## 1.147 ADDBOX

FUNCTION: add a Bevel Box to a window

TEMPLATE: WINDOW/N/K window to add the box to. If not specified, the current window will be used.  
 LEFTEDGE/N/A coordinates of the upper-left corner of the box  
 TOPEDGE/N/A  
 WIDTH/N/A size of the box  
 HEIGHT/N/A  
 TYPE/N/A see below  
 RECESSED/S

The RESULT variable is set to the number of the new box.

Available types of Bevel Boxes:

- 1 - Button
- 2 - Ridge (V39)
- 3 - IconDropBox (V39)

## 1.148 SETBOXATTR

FUNCTION: set some attributes of a bevel box.

TEMPLATE: WINDOW/N/K window of the box. If omitted, the current window will be used.  
 BOX/N/A number of the box.  
 L=LEFTEDGE/N/K  
 T=TOPEDGE/N/K  
 W=WIDTH/N/K  
 H=HEIGHT/N/K  
 TYPE/N/K see ADDBOX  
 RECESSED/S if specified, the text will \*toggled\* recessed.

## 1.149 ADDITEXT

FUNCTION: add an IntuiText in a window.

TEMPLATE: WINDOW/N/K window where the text should be added. If omitted the current window will be used.  
 FP=FRONTPEN/N/A  
 BP=BACKPEN/N/A  
 M=DRAWMODE/N/A see GETITEXTATTR

```

X=LEFTEDGE/N/A
Y=TOPEDGE/N/A
T=TEXT/A
FONT/N/K      number of the font to use. If not specified, the
               screen font will be used.

```

The number of the new text will be put into the RESULT variable.

## 1.150 GETITEXT

FUNCTION: asks the user to select an IntuiText.

```

TEMPLATE: WINDOW/N      number of the window to active. If omitted, the
                        current window will be used.

```

This function returns the error code 5 if the window didn't have any text or if the user aborted the command; a value of 10 is returned if the window was closed or if the window number given was out of range.

## 1.151 GETITEXTATTR

FUNCTION: fill a stem variable with info about an IntuiText.

```

TEMPLATE: WINDOW/N/K    itext's window - if not specified, the current
                        window will be used.
          TEXT/N/A      number of the text
          STEM/A        variable to fill

```

These are the fields filled:

```

stem.FRONTPEN      Pens of the text
stem.BACKPEN
stem.DRAWMODE      0 = JAM1, 1 = JAM2, 2 = COMPLEMENT
                  5 = JAM2 + INVERSVID
                  6 = COMPLEMENT + INVERSVID
stem.LEFTEDGE      Position of the text
stem.TOPEDGE
stem.FONT          number of the font. -1 if it uses the screen font.
stem.TEXT          the Text, finally... ;-)

```

## 1.152 GETITEXTLENGTH

FUNCTION: returns the length (in pixels) of an IntuiText.

```

TEMPLATE: WINDOW/N/K    itext's window - if not specified, the current
                        window will be used.
          TEXT/N/A      number of the text

```

## 1.153 SETITEXTATTR

FUNCTION: set some parameters of an IntuiText.

TEMPLATE: WINDOW/N/K        text's window. If omitted, the current window  
                                 will be used.  
         TEXT/N/A            number of the text.  
         FP=FRONTPEN/N/K  
         BP=BACKPEN/N/K  
         M=DRAWMODE/N/K  
         X=LEFTEDGE/N/K  
         Y=TOPEDGE/N/K  
         FONT/N/K  
         T=TEXT/K

## 1.154 GETDRIPEN

FUNCTION: return the color number associated to the specified pen.

TEMPLATE: PEN/N/A    one of the following:

- 0 - Detail Pen
- 1 - Block Pen
- 2 - Text Pen
- 3 - Shine Pen
- 4 - Shadow Pen
- 5 - Fill Pen
- 6 - Fill Text Pen
- 7 - Background Pen
- 8 - Highlight Text Pen
- 9 - Bar Detail Pen        (V39)
- 10 - Bar Block Pen        (V39)
- 11 - Bar Trim Pen        (V39)

See Screen/DriPens... for more informations about the pens.

## 1.155 GETBOX

FUNCTION: asks the user to select a Bevel Box.

TEMPLATE: WINDOW/N        number of the window to active. If omitted, the  
                                 current window will be used.

This function returns the error code 5 if the window didn't have any bevel box or if the user aborted the command; a value of 10 is returned if the window was closed or if the window number given was out of range.

## 1.156 GETBOXATTR

FUNCTION: fill a stem variable with info about a Bevel Box.

TEMPLATE: WINDOW/N/K box's window - if not specified, the current  
window will be used.  
BOX/N/A number of the box  
STEM/A variable to fill

These are the fields filled:

stem.LEFTEDGE	position
stem.TOPEDGE	
stem.WIDHT	size
stem.HEIGHT	
stem.TYPE	see ADDBOX
stem.RECESSED	boolean

## 1.157 GETFONTATTR

FUNCTION: fills a stem variable with infos about a font.

TEMPLATE: FONT/N/A number of the font  
STEM/A variable to fill

This are the fields filled:

stem.NAME	name of the font
stem.YSIZE	ta_YSize
stem.FLAGS	ta_Flags
stem.STYLE	ta_Style

This function returns the error code 10 if the font number is out of range.

## 1.158 GETSCRFONT

FUNCTION: fills a stem with infos about the screen font.

TEMPLATE: STEM/A variable to fill.

See GETFONTATTR for more informations about the fields filled.

## 1.159 GETFILE

FUNCTION: opens an Asl file requester

TEMPLATE: TITLE/A requester title

---

FILE/K	initial file
PATH/K	initial path
PATTERN/K	initial mask
SAVE/S	if set, the SaveMode will be turned on

This functions returns in RESULT the name of the file (with its path), or the error code 5 in RC if the user selects "Cancel" in the requester.

## 1.160 Index

### A

- ADDBOX
- ADDITEXT
- ARexx commands
- ARexx Interface Editor
- ARexx Routines
- ARexx scripts shipped with IEditor

### B

- Boolean Gadgets
- BUTTON kind

### C

- CHECKBOX kind
- Configuration required
- Copyright and Distribution
- Credits
- CYCLE kind

### D

- Disclaimer

### F

- Font Adaptivity
- Future plans

### G

- Gadget Banks
- Gadget Selection
- Gadget Tags
- Gadgets/Align/Bottom
- Gadgets/Align/Left
- Gadgets/Align/Right
- Gadgets/Align/Top

---

Gadgets/Clone/Both  
Gadgets/Clone/Height  
Gadgets/Clone/Width  
Gadgets/Copy  
Gadgets/Font...  
Gadgets/Items...  
Gadgets/Load...  
Gadgets/Remove  
Gadgets/Save...  
Gadgets/Select all  
Gadgets/Size...  
Gadgets/Spacing/Set X...  
Gadgets/Spacing/Set Y...  
Gadgets/Spread/Horizontally  
Gadgets/Spread/Vertically  
Gadgets/TabCycle order...  
Gadgets/Tags...  
GENERATEASM  
GENERATEC  
GENERATEE  
GETACTWDATA  
GETBOX  
GETBOXATTR  
GETDRIPEN  
GETFILE  
GETFONTATTR  
GETIMAGE  
GETIMAGEATTR  
GETITEXT  
GETITEXTATTR  
GETITEXTLENGTH  
GETNAME  
GETSCRFONT  
GETWNDDATA

H

How to register  
IDCMP Handler  
Key Handler

I

IEditor's author... err...hey, it's me! :)  
IEditor's Menus  
Images bank  
INTEGER kind  
Introduction

K

Keyboard commands

---

## L

LISTVIEW kind

## M

Macros/Add...

Macros/Execute...

Macros/Function keys...

Macros/Remove...

Main Routines

Making a Gadget

Menus

Menu Editor

Minimal configuration required

Misc/Localization...

Misc/main()...

Moving & Sizing Gadgets

MultiSelect, Listviews

MX kind

## N

NEW

Notes

NUMBER kind

## O

OPEN

## P

PALETTE kind

Preferences/Asm Sources...

Preferences/C Sources...

Preferences/Create Icons

Preferences/Current window in front of

Preferences/Generator...

Preferences/Save

Preferences/Tools Window

Preferences/Use Gadgets

Preferences/Use WFlags

Preferences/Workbench

Project/About...

Project/Exit

Project/Generate Source...

Project/New

Project/Open...

Project/Parameters...

Project/Save As...

Project/Save

---

## Q

QUIT

## S

SAVE

SAVEAS

Screen/DriPens...

Screen/Font...

Screen/Load...

Screen/Palette/Load...

Screen/Palette/Modify...

Screen/Palette/Save...

Screen/Save...

Screen/Tags...

Screen/Type...

SCROLLER kind

SETBOXATTR

SETIMAGEATTR

SETTEXTATTR

SETNAME

Shared IDCMP Port

SLIDER kind

Source

Starting with Interface Editor

STRING kind

## T

TEXT kind

Tools Window

## U

Updates

## W

Window/Discard All

Window/Discard

Window/Flags...

Window/IDCMP...

Window/Load...

Window/New...

Window/Open...

Window/Save...

Window/Tags...

Windows/BevelBoxes/Add

Windows/BevelBoxes/Edit...

Windows/Close All

Windows/Close



---

Windows/Images/Add  
Windows/Images/Delete  
Windows/Images/Move  
Windows/Print  
Windows/Size...  
Windows/Texts/Add...  
Windows/Texts/Delete...  
Windows/Texts/Edit...  
Windows/Texts/Move...  
Windows/Title...  
Windows/Zoom...

---