

GED.HYPER

COLLABORATORS

	TITLE : GED.HYPER		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 27, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GED.HYPER	1
1.1	main	1
1.2	FEATURE LIST	1
1.3	LICENCE	2
1.4	INTRODUCTION	3
1.5	REQUIRED SYSTEM	3
1.6	GETTING STARTED	4
1.7	DESCRIPTION OF MENUS	5
1.8	PROJECT MENU	6
1.9	project/about	6
1.10	project/user	6
1.11	project/clear text	7
1.12	project/more ed	7
1.13	project/load	7
1.14	project/load fast	8
1.15	project/load new	8
1.16	project/load original	8
1.17	project/insert	9
1.18	project/append	9
1.19	project/new name	9
1.20	project/current dir	9
1.21	project/save	10
1.22	project/save as	10
1.23	project/save as XPK	10
1.24	project/save & exit	10
1.25	project/bits	11
1.26	project/print	11
1.27	project/quit	11
1.28	BLOCK MENU	12
1.29	block/mark	12

1.30 block/hide mark	13
1.31 block/cut	13
1.32 block/copy	13
1.33 block/paste	13
1.34 block/bcopy	14
1.35 block/bdelete	14
1.36 block/bmove	14
1.37 block/delete column	14
1.38 block/insert column	14
1.39 block/append text	15
1.40 block/column text	15
1.41 block/indent	15
1.42 block/sort	16
1.43 block/uppercase	16
1.44 block/lowercase	16
1.45 block/save as	16
1.46 block/print	16
1.47 LAYOUT MENU	17
1.48 layout/borders	17
1.49 layout/word wrap	17
1.50 layout/block left/right	18
1.51 layout/block left	18
1.52 layout/block right	18
1.53 layout/block center	18
1.54 layout/section block	19
1.55 layout/section left	19
1.56 layout/section right	19
1.57 layout/section center	19
1.58 FIND MENU	20
1.59 find/find	20
1.60 find/find next	20
1.61 find/find previous	20
1.62 find/replace	21
1.63 find/replace next	21
1.64 find/count	21
1.65 find/check	21
1.66 find/matching bracket	22
1.67 find/reference	22
1.68 find/reference...	22

1.69 find/complete	23
1.70 find/ASCII table	23
1.71 find/ASCII insert	23
1.72 find/show code	24
1.73 find/character set	24
1.74 find/insert code FF	24
1.75 find/insert code ESC	24
1.76 find/functions	25
1.77 CONTROL MENU	25
1.78 control/go to line	26
1.79 control/top-bottom	26
1.80 control/to last change	26
1.81 control/store position	26
1.82 control/recall position	27
1.83 control/fold all	27
1.84 control/unfold all	28
1.85 control/insert-overwrite	28
1.86 control/toggle TAB mode	28
1.87 control/arrange windows	29
1.88 control/center window	29
1.89 control/zip window	29
1.90 control/enlarge window	29
1.91 control/freeze window	29
1.92 control/unfreeze window	30
1.93 control/next window	30
1.94 control/iconify	30
1.95 MISC MENU	30
1.96 misc/help	31
1.97 misc/project setup	31
1.98 misc/pick line	31
1.99 misc/push line	32
1.100misc/swap lines	32
1.101misc/double line	32
1.102misc/undo line	32
1.103misc/statistics	32
1.104misc/calculator	33
1.105misc/new shell	33
1.106misc/delete file	33
1.107misc/HiSpeed	33

1.108misc/insert date	34
1.109misc/insert time	34
1.110misc/command	34
1.111misc/last error	35
1.112MACRO MENU	35
1.113macros/edit macro	35
1.114macros/run text as macro	35
1.115macros/record start-stop	36
1.116macros/play	36
1.117macros/load	36
1.118macros/save	37
1.119macros/dmake	37
1.120macros/compile+link text	37
1.121macros/number	37
1.122macros/revision	38
1.123macros/GUIMake	38
1.124macros/config	39
1.125macros/options	39
1.126macros/modules	39
1.127macros/compile & link	39
1.128macros/compile & link new	40
1.129macros/display errors	40
1.130macros/error: first	40
1.131macros/error: previous	40
1.132macros/error: next	41
1.133macros/run executable	41
1.134CONFIG MENU	41
1.135config/keyboard bindings	42
1.136Event definition	42
1.137Magic codes	43
1.138config/TABs	43
1.139config/references	44
1.140config/dictionary	44
1.141config/indentation	45
1.142config/display	45
1.143config/menus	46
1.144config/printer	46
1.145config/misc	47
1.146config/calender-time	48

1.147config/save	48
1.148config/load	48
1.149Keyboard	49
1.150Cursor keys	49
1.151HELP key	50
1.152TAB key	50
1.153RETURN key	51
1.154DEL key	51
1.155ESC key	52
1.156F-Keys	52
1.157ARexx port	53
1.158Select a host	53
1.159Lock a window	54
1.160Do your Job	54
1.161Unlock GUI	55
1.162Internal commands	55
1.163Command list	56
1.164BACK	56
1.165BIND	56
1.166BITS	57
1.167BLOCK	57
1.168BRACKET	57
1.169CLIP	57
1.170CMD	58
1.171CODE	58
1.172COLON	58
1.173CONFIG	59
1.174CR	59
1.175DEL	59
1.176DELETE	59
1.177DIR	60
1.178DJUMP	60
1.179DOWN	60
1.180DPAGE	60
1.181ENDWORD	61
1.182EVENT	61
1.183EXALL	61
1.184FDOWN	62
1.185FILE	62

1.186FIND	62
1.187FIRST	63
1.188FOLD	63
1.189FORMAT	63
1.190FREEZE	63
1.191FUNC	64
1.192FUP	64
1.193GOTO	64
1.194HELP	65
1.195INDENT	65
1.196INFO	65
1.197INSERT	66
1.198KEY	66
1.199LAYOUT	66
1.200LEFT	67
1.201LINES	67
1.202LOCK	67
1.203MACRO	67
1.204MARK	68
1.205MAXDOWN	68
1.206MAXUP	68
1.207MENUS	68
1.208MISC	69
1.209MODE	69
1.210MORE	69
1.211NAME	70
1.212NEW	70
1.213NEXT	70
1.214OPEN	70
1.215PATH	71
1.216PHRASE	71
1.217PING	71
1.218PONG	72
1.219POP	72
1.220PREFS	72
1.221PREV	72
1.222PREVEND	73
1.223PRINT	73
1.224PROJECT	73

1.225PUSH	74
1.226QUERY	74
1.227QUIT	76
1.228REFRESH	76
1.229REMAP	76
1.230REPLACE	77
1.231REQUEST	77
1.232RIGHT	78
1.233RX	78
1.234SAVE	78
1.235SCREEN	79
1.236SHIFT	79
1.237SMARTCR	79
1.238TAB	80
1.239TABS	80
1.240TASK	80
1.241TEXT	80
1.242UJUMP	81
1.243UNDO	81
1.244UNLOCK	81
1.245UP	82
1.246UPAGE	82
1.247USE	82
1.248VIEW	83
1.249WINDOW	83
1.250XREF	83
1.251GENERAL HINTS	84
1.252CREDITS	85
1.253HOW TO REGISTER	85
1.254HOW TO GET UPDATES	86
1.255HOW TO CONTACT AUTHOR	86
1.256 GoldED	87

Chapter 1

GED.HYPER

1.1 main

Feature list	Command list
Licence	General hints
Introduction	Credits
Required system	How to register
Getting started	How to get updates
Description of menus	How to contact author
Keyboard	
ARexx port	???

Suggested reading is from top to bottom. Novice users should skip the section on GoldED's internal commands respectively its ARexx interface.

1.2 FEATURE LIST

FEATURE LIST

- o OS3.x look & functions (AppWindows & more)
 - o Folding of paragraphes (unlimited nesting available)
 - o Menus fully customizable (easy-to-use requesters)
 - o ARexx port (about 250 commands/options)
 - o QuickReference capability (includes, sources, ...)
 - o APC (TM): Automatic phrase completion, based on dictionary
 - o shifting (two speeds)
 - o Smooth display, fast scrolling
 - o HotKey support
 - o unlimited number of windows
 - o Formatter (aligned, block, centered), WordWrap
 - o automatic indentation (left margin)
 - o SmartIndention (code dependend, e.g. after 'if')
 - o AutoBackup (any interval, any backup path)
 - o MenuHelp (AmigaGuide)
 - o localized (English/German so far)
 - o printer control (linefeed/spacing/style)
-

- o Multiselect of files (e.g. join files together)
- o direkt XPK support : (de)crunching
- o ASCII character selection window
- o flexible GUI handling: any display mode / font
- o two window modes: normal vs. borderless windows
- o protection bits support, file comment support
- o clipboard support (snapping): Cut & paste
- o AutoLoad of project file
- o FastLoad mode
- o AutoCase (TM) correction (based on user's dictionary)
- o automatic parenthesis check available
- o QuickFunc jump table display for many prog. languages
- o insertion of columns , removal of columns
- o user friendly (about 25 requesters)
- o fixed/regular/ dynamic TABs ; solid/transparent TABs
- o character set remap (e.g. Amiga to MS-DOS)
- o four speeds of scrolling
- o quick starter ED
- o startup macro
- o powerful macro recording
- o and many more ...

1.3 LICENCE

LICENCE

This licence agreement affects all programs, manuals and disks of the editor package GoldED (registered or unregistered) except files within the support directory. Violations of any of the topics below will cause this licence to expire, i.e. terminate your rights to use or otherwise handle (e.g. distribute) the package.

You may - apart from backups for private use - neither copy this program, nor modify it nor transfer it to other media. This doesn't apply for the demo version. You may duplicate/distribute the demo version as long as you don't charge money for it apart from compensation for time/media expenditures. You may not distribute the registered version.

You are not allowed to include the demo version into any kind of software collection apart from the Amiga Library (compiled by Fred Fish) and BBS file areas without written permission of the author. This does apply for CD roms, disk magazines, ... The demo version may not be distributed as part of a bundle (e.g. bookware, disk magazine).

The copyright owner(s) reserve the right to interdict distribution at any time if the distributor fails to obey to this licence from the copyright owner(s) point of view. Profits made by illegal distribution have to be passed to the copyright owner(s).

You may not modify the package: do not change the compilation, do not change (patch) files. The editor might become inoperational if you do so.

The copyright owner(s) provide this program "as is". The entire risk of using

this program is with you. The copyright owner(s) won't ever be reliable for damages to you - whether they arise from the use of this package or the inability to use it, whether they are direct or consequential, including, but not limited to, the loss of data or the generation of inaccurate data.

1.4 INTRODUCTION

INTRODUCTION

The basic design goals of this editor were: make it as user friendly as possible. And make it as fast as possible. A lot of attention has been put to the general look & feel, performance considerations and full support of the great Amiga OS. The whole design is based on OS2.1/3.0, a fast CPU, sufficient RAM and a hard disk. I neither cared much about memory consumption nor did I pay attention to OS1.3 compatibility since this would have meant to make concessions to speed or general handling.

GoldED's way of working is basically event based: the editor does wait for events like a pressed key or a menu selection and then calls a dispatcher to perform appropriate action. Action is not hard coded: you may assign any of the editor's internal functions to an event (though most people won't do any assignments at all but use the default configuration). For example you may assign the 'open file' function to the 'A' key by using GoldED's keyboard requester. Or assign the text "hi, I'm the A key" to the A key. Or assign an ARexx macro script to it. Or a DOS command. Or just leave it as it is. No matter what kind of interface to GoldED you use (the menus, the keyboard or its ARexx port): all of them do support the same command set. It's easy to use, straight-forward and flexible. GoldED doesn't use an internal macro language like DME: Its internal functions are basically function calls, performing some kind of action. It fully relies on ARexx as far as conditional statements like IF ... THEN are concerned.

Please keep in mind that this is the very first (beta) release of GoldED. Some features (especially the ARexx port) require additional work. I do expect bugs to show up. The settings file format as well as the command set of GoldED's ARexx interface may change with the first official release.

1.5 REQUIRED SYSTEM

REQUIRED SYSTEM

Minimum requirements are OS2.04, 68000 and 1 MB RAM. GoldED has not been tested with a 68000. Suggested system for average performance (I'm serious about that) is OS3.0, 68020 & 2 MB RAM. Sounds like the A1200, doesn't it? Some features are not available under OS2.04. I strongly recommend to install/rekick OS3.x if your Amiga is capable of it. GoldED is *much* faster under OS3.x.

Software requirements (all these files have been released to the public already - check Fish disks):

- o regtools library
- o xpk library
- o amigaguide library

1.6 GETTING STARTED

GETTING STARTED

Start the editor either by doubleclicking its icon or by typing its name at shell level. Or use the HotKey combination (i.e. right ALT + right SHIFT & RETURN) if the editor is present in the background already. Depending on whether or not AutoLoad (see config/misc) is enabled, it might happen that a file is loaded automatically even though you have provided no file name. Don't get confused - it's a feature, not a bug :-)

QuickStarter

Besides the main editor GoldED, a quick starter ED is provided. This program is a small (4 KB) GoldED frontend. You can use it AS IF it were a real editor. For example you might type 'ed letter' to edit the file letter if the quick starter were named 'ed'. The big advantage of a quick starter is its ability to pass a new job to an already running instance of GoldED (if there is any). Needless to say that this happens very fast. The quick starter may be made resident (the main editor may not). The source code of ED is available in the "GoldED:Tools/EDSource" drawer.

Arguments

GoldED accepts four kinds of arguments: a list of file names to load, the name of a public screen to open on (after the SCREEN keyword), a configuration file to use (CONFIG keyword) and finally the HIDE option. The SCREEN/CONFIG/HIDE arguments may be passed as command line options as well as icon tool types (don't use quotes around file names within tool type entries). Example:

```
GoldED s:startup-sequence CONFIG s:MyPrefs
```

If you want the editor to stay in the background initially (waiting for HotKey activation), use the HIDE option but don't supply file names:

```
GoldED HIDE
```

The quickstarter ED additionally supports the STICKY option (see below). It will ignore the CONFIG/SCREEN options if it is able to pass the list of files to a running instance of GoldED. Example:

```
ED mail:answer CONFIG s:BBS.prefs STICKY
```

The quickstarter has been designed to for synchronous operation (except if the HIDE option is used to run the editor in the background). It won't detach itself from a shell window unless you RUN it. If the STICKY option is not specified, a call to ED returns after the last GoldED window has been closed. If STICKY is specified, a call to ED returns after the window opened by ED is

closed (GoldED itself may keep on running if there are further windows to handle). This is designed to be used in command files where you want the execution of the batch file or command script to wait until the user has finished editing a particular file. ED will return (using the sticky option) when the user quits out of the file.

MenuHelp

The editor's help facility is based on Commodore's AmigaGuide library. Simply doubleclick the 'GoldED.guide' icon - its default tool is AmigaGuide. You might use 'MultiView' of OS3.0 to read the 'GoldED.guide' file, too. GoldED itself does support menu help: press the <HELP> key during selection of a menu item (within GoldED) to get explanations related to the item you selected.

Language

Locale library of OS2.1 or better is supported as far as GoldED's requesters are concerned: if you have selected German to be your default language (by using the 'locale' preferences of your Workbench), requesters as well as time/date strings will be German. So far only English and German are available; other languages might be available in the future. Locale settings do not affect menus because menus are not part of the editor itself but external text files. Use config/menus to edit/load menus. The default menu definition file is available in German & English. It is installed by the Install utility when installing the GoldED package. You may load a new menu definition file at any time (see config/menus).

GUI

The editor's user interface does support the look & feel of OS2.x. Most gadgets offer keyboard activation: if a character of the gadget inscription appears underscored, it may be used as shortcut. Thus you may move a slider gadget or activate a button gadget by a single keystroke. Use the shift key simultaneously to toggle 'direction' of changes - e.g. to move a slider gadget one step to the left instead of to the right. Use the 'Amiga' key simultaneously to activate another gadget from within a string gadget (without R-Amiga your input would go to the string gadget).

1.7 DESCRIPTION OF MENUS

DESCRIPTION OF MENUS

Since the editor does offer many possibilities for user configuration, there is no 'standard' appearance, neither as far as colors, resolution or fonts nor as far as menus & functions are concerned. This manual does describe the default configuration. Use config/load to load a settings file.

The standard menus are:

Project menu	Control menu
Block menu	Misc menu
Layout menu	Macro menu
Find menu	Config menu

1.8 PROJECT MENU

PROJECT MENU
menu tree of project menu

project/about	project/new name
project/user	project/current dir
project/clear text	project/save
project/more ed	project/save as
project/load	project/save as XPK
project/load fast	project/save & exit
project/load new	project/bits
project/load original	project/print
project/insert	project/quit
project/append	

The project menu offers a variety of commands related to general handling of a document, especially as far as basic input/output functions are concerned (load/save or printing).

1.9 project/about

project/about of PROJECT MENU

Displays version ID. Furthermore the name of this task's ARexx port and this task's screen name are displayed. If you told the editor to open a custom screen, this screen is always 'public', i.e. you are invited to run other tasks on the editor's screen (example: type SHELL CON:0/11/640/100/Shell/screenGOLDED.1 to open a shell window on the editor's screen). See config/display for information on how to make other programs appear on GoldED's screen.

1.10 project/user

project/user of: PROJECT MENU

If you are a registered user of GoldED, your name would be displayed when you select this menu item. The unregistered release does show a general copyright information.

1.11 project/clear text

project/clear text of: PROJECT MENU

Clears contents of current window. Protection bits are set to the defaults (see config/misc) before any further action takes place. The path is reset to the current path, the file name is reset to "unnamed". You are asked for confirmation if the window's text has been changed since loading

1.12 project/more ed

project/more ed of: PROJECT MENU

Does open further window for input. The window size is read from the configuration file (see config/load). If you have enabled AutoCenter (see config/misc , default is ON), the new window is centered on screen, otherwise the window position is read from the configuration file, too. Use config/save to save window dimensions & position of your current window as default settings. This editor does offer many functions related to window management, e.g. it is able to 'arrange' your windows on screens. Autoscroll screens are fully supported: window functions consider the visible section only. See control/arrange windows for more details.

1.13 project/load

project/load of: PROJECT MENU

Discards current text if any was loaded (internally realized as project/clear text), asks for new file(s) & loads the file(s).

Multiselect

Like most file requesters of GoldED, the file requester used by this function does support multi select, i.e. you may select more than one file at once. Treatment of multi selected files depends on the context: while this function will open a new window for each file, project/append will load all selected files to one window.

AppWindows

GoldED's text windows are so called AppWindows: it is possible to drag icons (of text files) over a text window. These files would be appended to the text of that window. Multi select is supported: You may drag more than one file over a window using extended selection (hold down the shift key while you select icons).

Crunched XPK files (see project/save as XPK) are recognized and decompressed while reading if the XPK library is available. This function does check for TABs (dez. 8); TABs will be replaced by SPC (dez. 32) while loading. See config/tabs if you want to influence substitution. LOAD does

use a reasonable sized input buffer of about 16KB to speed up operation. However, if you want even better performance, use `project/load fast` instead.

If `AutoFold (config/misc)` is ON, the file is scanned for folded sections automatically after loading. You should disable `AutoFold` if you don't want to use this editor's folding capability to prevent scanning for fold markers and thus save time.

Warning: Do never attempt to edit binaries (programs). GoldED is a text editor, not a file monitor. It will change the data in a way suitable for text files but definitely unsuitable for binaries (e.g. remove CR return codes, substitute spaces, clear the executable-bit).

1.14 `project/load fast`

`project/load fast of: PROJECT MENU`

`FastLoad`

A fast replacement for `project/load`. This option does require an IO buffer of the original file's size and it does not check for TABs. However, since GoldED does never write TABs (a concession to speed), this function might suit you very well if your Amiga has a decent amount of RAM. This function is about 50% faster than 'slow' load. Use `project/load original` to reload a file in slow mode (with TAB substitution enabled) if you discover after loading that it does contain TAB's ('strange' characters at beginning of lines). GoldED does automatically fall back to slow load if it detects TAB codes within the first 1500 bytes of a file.

1.15 `project/load new`

`project/load new of: PROJECT MENU`

Load a document but don't load it to the current window as `project/load` would do. Instead, a new window is prepared.

1.16 `project/load original`

`project/load original of: PROJECT MENU`

Reload current file from disk. Useful after you have made some changes but want to switch back to the original. This function does use 'slow loading' (see `project/load fast`), i.e. TAB's will be substituted by spaces.

1.17 project/insert

project/insert of: PROJECT MENU

Insert a file at current cursor position (before current line). A file requester will pop up, asking you for one or more files to be inserted. Hold down the SHIFT key to select more than one file (this feature is called 'multiselect').

1.18 project/append

project/append of: PROJECT MENU

Append one or more files to your current text. A file requester will pop up, asking you for one or more files to append to your text. This function is quite useful to join a couple of files together.

1.19 project/new name

project/new name

Useful to change the name of current text (you are prompted for a new one). Only the document in memory is affected - no disk file is renamed. Since one usually would use PROJECT/SAVE AS to save a file to a new location, this function is rarely used.

1.20 project/current dir

project/current dir of: PROJECT MENU

This function sets the 'current path' to wherever you want. The current path is used by many functions (e.g. project/load new) as default path. It is used by all menu items of type DOS, too (e.g. 'new shell'). If you have set the current path to 's:', a function like project/load new would list the s-directory when asking for a file.

Setting the current path doesn't change the name of a text - use the menu command project/new name to do that. Some functions (like project/load) don't care about the current path - they extract path information from the current document's name. Use project/clear text to reset the path of an empty text window.

1.21 project/save

project/save of: PROJECT MENU

Save your current text, using the current name displayed in/below the window's title bar. Any old copy (no matter how protection bits have been set) is overwritten without asking for confirmation. However, you might turn on backup creation if you want the editor to backup the old version if any is available; see config/misc for more information on backups. You might even ask the editor to backup your files regularly (e.g. every 10 minutes) to any directory (see config/misc as well).

This function is disabled for windows of type read-only to prevent the user from accidentally overwriting important files. QickRef (TM) windows are read-only (see QuickReference).

1.22 project/save as

project/save as of: PROJECT MENU

Same as project/save but does give you the opportunity to enter a new file name before the text is written to disk.

This function is disabled for windows of type read-only to prevent the user from accidentally overwriting important files. QickRef (TM) windows are read-only (see QuickReference).

1.23 project/save as XPK

project/save as XPK of: PROJECT MENU

XPK support

Save current file in compressed mode (about 50% less disk space required). See config/misc on how to select compression mode. This function requires the complete set of XPK libraries which has been released as FD. Note: other editors/programs might not be able to handle crunched files, so be careful when using this function. Don't crunch your sources - the compiler won't like it.

1.24 project/save & exit

project/save & exit of: PROJECT MENU

Same as project/save followed by project/quit : Save current text & close window. Exit from GoldED if the last window has been closed. GoldED won't

close a window respectively exit if the SAVE operation fails (e.g. disk full error).

This function is disabled for windows of type read-only to prevent the user from accidentally overwriting important files. QickRef (TM) windows are read-only (see QuickReference).

1.25 project/bits

project/bits of: PROJECT MENU

Edit protection bits of the current text. Have a look at your Amiga DOS manual if you are unfamiliar with these bits. In generally you should set the S (script) bit for batch files but let the other bits untouched. Changes won't have any effect until you save the text. These bits are set to a default state after performing project/clear text). Use config/misc to define the default state.

1.26 project/print

project/print of: PROJECT MENU

Send current text to printer 'PRT:'. See config/printer for information on how to affect output style. DeskJet/LaserJet owners should use the HiSpeed printing facility of the misc menu instead (misc/hispeed). HiSpeed is shareware; a registered HiSpeed release is part of the GoldED Pro registration.

1.27 project/quit

project/quit of: PROJECT MENU

Close current window. Exit from GoldED if the last window has been closed. You are asked for confirmation if you attempt to exit without having saved your text so far (unless you didn't change the text at all). You are NOT asked for confirmation if only preferences have been changed (see config/save).

GoldED's memory management is asynchronous - after having closed a window you don't have to wait for the memory to be freed (this is done by a background task). Your Amiga might appear to be slightly slower than usual while the background task is busy - especially if the text buffer has been large (200 KB or more).

HotKey

If hotkey support (see config/misc) is enabled, GoldED will not be removed

from RAM even after closing the last window. Instead it will wait for a hotkey combination (right ALT & right SHIFT & RETURN). Press these keys to make the editor reappear. Or use the commodities exchange program of your workbench. HotKey activation will give you a very fast response time since the editor won't have to be reloaded from disk.

1.28 BLOCK MENU

BLOCK MENU
menu tree of block menu

block/mark	block/insert column
block/hide mark	block/append text
block/cut	block/column text
block/copy	block/indent
block/paste	block/sort
block/bcopy	block/uppercase
block/bdelete	block/lowercase
block/bmove	block/save as
block/delete column	block/print

All functions of the block menu are related to the management of 'blocks', which are marked sections of lines. Ususally one would use menu functions (respectively corresponding keyboard shortcuts) to mark lines, however the mouse may be used as well: simply hold down the left mouse button while you drag the pointer over a desired section of lines. You can have only one block per document. Blocks work on a line-by-line basis, i.e. it is not possible to mark single words. This will change with future releases of GoldED.

1.29 block/mark

block/mark of: BLOCK MENU

Mark beginning or end of a block. If you call this function for the first time (no marked lines so far), the editor will remember the current line as START of a new block. If you call this function for a second time, the editor considers the current cursor position as the block's END - all lines between START and END are highlighted. START respectively END are symbolic line names - START doesn't necessarily have to be smaller than END. Things are handled differently if you use this function after a block has already been selected: If cursor position is closer to the end of the current block than to its start, the end position is updated. Otherwise the start position is set to the cursor's line. You might use block/hide mark to get rid of a block, i.e. to unmark lines.

1.30 block/hide mark

block/hide mark of: BLOCK MENU

Turn block off, ie. don't have any text section highlighted. Useful after block/mark to get rid of a block selection.

1.31 block/cut

block/cut of: BLOCK MENU

Cut selected lines (see block/mark) from text. These lines are written to the clipboard, ready to be inserted into any application supporting the clipboard device (e.g. into a shell window by pressing AMIGA & V or into any GoldED document using block/paste). You shouldn't use these clipboard-based functions to move/copy data within a single GoldED document; block/bcopy or block/bmove would perform the same task much faster.

1.32 block/copy

block/copy of: BLOCK MENU

Copy marked lines (see block/mark) to the clipboard device, ready to be inserted into any application supporting this device. This function does not affect your current text (as block/cut would). See block/paste for more details.

1.33 block/paste

block/paste of: BLOCK MENU

Insert clipboard contents (if any are available) into current text.

Clipboard

The 'clipboard' (actually the 'CLIPS:'-directory) is used by many applications to exchange data. Postings to the clipboard are IFF files. GoldED does support IFF/FTXT clipboard access. For example, you could mark a text within a shell window, press AMIGA + C to copy these lines to the clipboard and then reinsert them into a GoldED window using block/paste . A paste operation won't remove the data from the clipboard, i.e. you can call this function several times.

1.34 block/bcopy

block/bcopy of: BLOCK MENU

Copy marked lines (see block/mark) to current cursor position. Useful to duplicate sections within a single document (while you would have to use a block/cut / block/paste pair to exchange data between different windows or different applications).

1.35 block/bdelete

block/bdelete of: BLOCK MENU

Delete highlighted lines (see block/mark). It is not possible to undo this operations, so better be careful. If you don't want to miss an undo option, use block/cut instead (block/paste to undo) - however those functions are no as fast as this one.

1.36 block/bmove

block/bmove of: BLOCK MENU

Move highlighted lines (see block/mark) from their original position to current cursor position. This function is useful to move sections of lines within a single document (while you would have to use a block/cut / block/paste pair to move text from one window to another window/application).

1.37 block/delete column

block/delete column of: BLOCK MENU

Removal of columns

Delete a column from highlighted lines (see block/mark). Move cursor to desired column before you call this function (e.g. move cursor to column 10 if you want to delete this column from all block line). You shouldn't use this function if you simply want to change indention of a paragraphe because - use block/indent instead.

1.38 block/insert column

block/insert column of: BLOCK MENU

Insertion of columns

Insert a column into highlighted lines (see block/mark on how to mark lines). Move cursor to desired column before you call this function. Example: Move cursor to column 40 if you want to insert one space character before this column into all marked lines.

1.39 block/append text

block/append text of: BLOCK MENU

Append text to marked lines. Example usage: Mark some lines (see block/mark) and call this function. A requester will pop up, asking you for a text to be appended. If you enter ';', a semicolon would be appended to each line you have marked.

1.40 block/column text

block/column text of: BLOCK MENU

Insert a text into marked lines at current cursor position. Example usage: Mark some lines (see block/mark), move cursor to desired column (e.g. column 1) and call this function. A requester will pop up, asking you for a text to be inserted. If you enter 'Prototype ', this word would be inserted at the beginning (column 1) of each block line. This function is quite useful if you want to create tables (for example you could insert a '|' to get a vertical line).

1.41 block/indent

block/indent of: BLOCK MENU

Shifting

Change indention of marked lines (block/mark). Use arrow gadgets to shift text left or right. Currently selected keyboard-TAB distance (see config/tabs) is used as default indention step, however you may change this value using the step gadget. An indent function (two speeds) is assigned to keyboard's cursor keys, too (see cursor keys).

1.42 block/sort

block/sort of: BLOCK MENU

Sort selected lines (see block/mark) alphabetically. This function is not case sensitive (i.e. 'A' and 'a' would be considered equal).

1.43 block/uppercase

block/uppercase of: BLOCK MENU

Make all characters of highlighted lines (see block/mark) uppercase. This function does use locale library if available to treat non-ASCII characters (e.g. 'ß') the right way. Locale library is part of OS2.1 and OS3.0 or better. It is not part of OS2.04.

1.44 block/lowercase

block/lowercase of: BLOCK MENU

Make all characters of highlighted lines (see block/mark) lowercase. This function does use locale library if available to treat non-ASCII characters (e.g. 'ß') the right way. Locale library is part of OS2.1 and OS3.0 or better. It is not part of OS2.04.

1.45 block/save as

block/save as of: BLOCK MENU

Save marked lines (see block/mark) to disk. You will be asked for a file name. Do not use this function to move text from one text to another - use the clipboard instead (see block/copy).

1.46 block/print

block/print of: BLOCK MENU

Print marked lines (see block/mark). The standard preferences printer is used. See config/printer if you want to affect output settings like quality or linefeed.

1.47 LAYOUT MENU

LAYOUT MENU
menu tree of layout menu

layout/borders	layout/block center
layout/word wrap	layout/section block
layout/block left/right	layout/section left
layout/block left	layout/section right
layout/block right	layout/section center

Formatter

All functions of the layout menu are related to formatting a text. They are of no use when writing source code – which is the main purpose of GoldED. But they should turn out useful if you edit normal text files (e.g. your e-mail).

Paragraphe vs. block

Some of the formatting functions actually are block functions: they do affect marked lines only. See `block/mark` if you don't know how to mark lines. Others do affect the 'current paragraphe' of a text. For example you might move the cursor to this star '*' and then choose `layout/section left`: the lines from 'Some ...' (beginning of paragraphe) to the end of this paragraphe would be made left aligned. The editor does determine the end of a paragraphe by looking for an empty line (however some lines appear to be empty but actually contain multiple spaces – these lines are not considered empty).

1.48 layout/borders

layout/borders of: LAYOUT MENU

Set layout area for formatting functions. 'Left border' as well as 'width' may be set. Example: Set border to 5 respectively width to 70 before using `layout/section block`. Your text be left & right aligned afterwards. Total width (including border) will be 75 characters, 'block' width 70, left margin 5 columns.

1.49 layout/word wrap

layout/word wrap of: LAYOUT MENU

WordWrap

Toggles word warp mode on/off. Current status is displayed in screen's title bar (either WRAP or NORM). If word wrap is on, the editor will reformat the current paragraphe (left aligned) if cursor moves behind right margin (see `layout/borders` on how to set right margin). Word wrap is very useful if you

work on a plain ASCII text: You won't have to bother about pressing the enter key - the editor will switch too a new line automatically if the current lines is full.

Do never use word wrap if you work on a source file - the source would get mixed up as soon as the editor attempts to reformat a section of lines. Do not use word wrap when creating tables or any other kind of formatted output for the same reason, too.

1.50 layout/block left/right

layout/block left/right of: LAYOUT MENU

Reformats marked lines (see block/mark) to make them appear left & right aligned. The block's last line is made left aligned. Empty lines are not removed during formatting, i.e. your document's basic structure remains unchanged. Multiple spaces are removed, so better do not attempt to format tables. Use layout/borders to set block width respectively left border.

1.51 layout/block left

layout/block left of: LAYOUT MENU

Reformats marked lines (see block/mark) to make them appear left aligned. Empty lines are not removed during formatting, i.e. your document's basic structure remains unchanged. Multiple SPC's are removed, so better do not attempt to format tables. Use layout/borders to set maximum line width respectively left border.

1.52 layout/block right

layout/block right of: LAYOUT MENU

Reformats marked lines to make them appear left aligned. Empty lines are not removed during formatting, i.e. your document's basic structure of paragraphes remains unchanged. Multiple SPC's are removed, so better do not attempt to format tables. Use layout/borders to set maximum line width respectively left border.

1.53 layout/block center

layout/block center of: LAYOUT MENU

Reformats marked lines to make them appear centered within currently selected

layout area. Use layout/borders to set width of layout area respectively its left margin. Empty lines are not removed during formatting, i.e. your document's basic structure of paragraphes remains unchanged.

1.54 layout/section block

layout/section block of: LAYOUT MENU

Reformats current paragraphe (see Paragraphe vs. block) to make the lines appear left & right aligned. The last line of a paragraphe is not affected - it is made left aligned. A line is a 'last' line if its successor is an empty line (however some lines appear to be empty but actually contain spaces - these lines are NOT considered empty). Empty lines are not removed during formatting, i.e. your document's basic structure of paragraphes remains unchanged. Multiple SPC's are removed, so better do not attempt to format tables. Use layout/borders to set block width respectively left border.

1.55 layout/section left

layout/section left of: LAYOUT MENU

Reformats current paragraphe (see Paragraphe vs. block) to make the lines appear left aligned. Empty lines are not removed during formatting, i.e. your document's basic structure remains unchanged. Multiple SPC's are removed. Use layout/borders to set layout width respectively left border.

1.56 layout/section right

layout/section right of: LAYOUT MENU

Reformats current paragraphe (see Paragraphe vs. block) to make the lines appear right aligned. Empty lines are not removed during formatting, i.e. your document's basic structure remains unchanged. Multiple SPC's are removed. Use layout/borders to set layout width respectively left border.

1.57 layout/section center

layout/section center of: LAYOUT MENU

Reformats current paragraphe (see Paragraphe vs. block) to make the lines appear centered within currently selected layout area. Use layout/borders to set layout width width respectively left border. Empty lines are not removed during formatting, i.e. your document's basic structure remains

unchanged. Multiple SPC's are removed.

1.58 FIND MENU

FIND MENU
menu tree of find menu

find/find	find/reference...
find/find next	find/complete
find/find previous	find/ASCII table
find/replace	find/ASCII insert
find/replace next	find/show code
find/count	find/character set
find/check	find/insert code FF
find/matching bracket	find/insert code ESC
find/reference	find/functions

1.59 find/find

find/find of: FIND MENU

Shows a requester to enter the text to search for. You may choose the search to be case sensitive or not by using the appropriate checkmark gadget. Use the NEXT gadget to go to the next (i.e. after current cursor position) occurrence of the text to search for. Use the FIND gadget to look for the first occurrence of the search pattern.

Note: case-sensitive search is much faster than case-insensitive search.

1.60 find/find next

find/find next of: FIND MENU

Go to the next occurrence (i.e. after current cursor position) of the pattern to search for. Use find/find to enter search text.

1.61 find/find previous

find/find previous of: FIND MENU

Go to the previous occurrence (i.e. before current cursor position) of the

pattern to search for. Use `find/find` to enter the search text.

1.62 find/replace

find/replace of: `FIND MENU`

Shows a requester to enter both, the text to search for as well as a replacement. You may choose the operation to be case sensitive or not by using the appropriate checkmark gadget. Select the `NEXT` gadget to replace the next occurrence of the search text by the replace text. Use the `ALL` gadget to replace all occurrences of the search text. Decide for the `BLOCK` gadget if you want the replacement operation to be restricted to marked lines (see `block/mark`).

1.63 find/replace next

find/replace next of: `FIND MENU`

Replaces next occurrence of search pattern by replace text. Replacement as well as the search pattern are set using the `find/replace` requester.

1.64 find/count

find/count of: `FIND MENU`

Shows a requester to enter a search text. You may choose the search operation to be case sensitive or not by using the appropriate checkmark gadget. Finally select the `COUNT` gadget to count all occurrences of the search pattern within your text.

1.65 find/check

find/check of: `FIND MENU`

Checks for correct use of braces `'(` within current line. You will be warned if there are more opening braces than closing ones or vice versa. Nesting is checked, too. You may turn on automatic checking after each line using `config/dictionary` .

1.66 find/matching bracket

find/matching bracket of: FIND MENU

Move cursor to matching bracket. Handles ()<>... depending on character under cursor. Useful to check levels of execution within a program (move cursor over first opening parenthesis within a C-function, then use this function).

1.67 find/reference

find/reference of: FIND MENU

QuickReference

Does try to open a help text related to the word your cursor is placed over. Example usage: Type 'struct RastPort', move cursor over 'RastPort', then call this function. If the reference system is set up (see below), a new text window should pop up, showing you a file 'graphics.h' from you compiler's include directory. The cursor will be placed in the first line of the structure definition 'RastPort'. The reference file is read-only, i.e. you may change its contents but you may not save it (this is to prevent you from accidentally overwriting important files).

Setting up the reference system

In order to have this command work properly you'll have to set up GoldED's reference system first (i.e. tell the editor where to look for files like 'graphics.h'): Use config/references to do this. Setting up the reference system basically consists of selecting the files or directories to be referenced (e.g. your includes directory); it's a matter of a few seconds. GoldED is shipped without the reference system set up.

What files can be referenced ?

It is possible to reference a lot of different file types like source codes (C, BASIC, Pascal, Assembler), autodocs or C-header files (*.h). See config/references for details. Example: you might set up the reference system to know the functions of your current programming project (probably consisting of many files). After having done this you could move the cursor over a function call of one of your own functions and then use find/reference. A new window would pop up, presenting you the lines of a file where this function is defined.

1.68 find/reference...

find/reference... of: FIND MENU

Prompts for a string to be referenced. Example usage (if the reference system is set up): Enter 'Window', then use OK. A new window would pop up, presenting you the header file from your compilers includes directory where a

structure 'Window' is defined. See find/reference for further explanations.

1.69 find/complete

find/complete of: FIND MENU

APC (Automatic Phrase Completion)

Trys to 'complete' the word your cursor is placed over. Example usage: type 'swin', then call this function. 'swin' would be replaced by 'struct Window' if the C-dictionary is present (see config/dictionary on how to load/edit/create a dictionary). You might type 'swindow' as well - it would be replaced by 'struct Window', too. You might even type 'swdow'. Or 'struwi'. However, something like 'wind' wouldn't be recognized since the abbreviation's first letter must always be the same as the first letter of the full form. This function uses the dictionary facility; the larger your dictionary grows the more detailed your abbreviations will have to be to ensure unique identification. A keyboard shortcut of this function is assigned to the ESC key .

1.70 find/ASCII table

find/ASCII table of: FIND MENU

Opens a character selection requester: All characters of your current font are displayed in a table. Pick the character you are looking for; it will be inserted at current cursor postion.

1.71 find/ASCII insert

find/ASCII insert of: FIND MENU

Prompts you for an ASCII code. Enter desired number (e.g. 65), then press return. The character will be inserted at current cursor position. The ASCII insert function is useful to embed control codes into your text. Example usage: Insert the codes 27 91 49 109 (calling this function 4 times) at the top of a document. This sequence is recognized by the printer device as 'turn bold on' command. The printer would switch to bold mode if the text were sent to the printer device (using project/print).

Often-used sequences should be assigned to menu entries instead of using this requester; see config/menus on how to create menu entries of type 'text'.

1.72 find/show code

find/show code of: FIND MENU

Show ASCII code of character under cursor. This might be useful to identify 'garbage' characters. Example usage: After loading of a large text file in fast mode (see project/load fast ; TAB substitution turned off) you discover strange characters at the beginning of some lines. Using this function you might find out that those characters actually are TAB codes (ASCII code 12). You then would have to reload (see project/load original) the text to have TABs replaced by spaces (use find/character set to get rid of other 'strange' codes).

1.73 find/character set

find/character set of: FIND MENU

Character set remap

Asks you for a character set translation file before GoldED attempts to remap the current text. Remapping means that each character is replaced by another character defined in the translation file. The result solely depends on the translation file. If you load the 'AmigaToMSDOS' file using the REQ(uester) gadget, the text would be remaped in a way to make it readable by MSDOS machines. You could use the 'MSDOSToAmiga' to do it vice versa (make MSDOS files readable for the Amiga). Or load 'StripControl' to have non-printable 'garbage' characters removed. Finally StripNonASCII is useful to remove non-ASCII characters (many e-mail networks do not allow non-ASCII codes).

Most translation files are 'lossy': some characters won't get translated. For example MSDOS doesn't know the '©' character, thus a '©' couldn't be remapped properly by an AmigaToMSDOS translation file (a 'c' would be used instead).

1.74 find/insert code FF

find/insert code FF of: FIND MENU

Inserts a form feed code (ASCII 12) at current cursor position. This code is recognized by printers. If the text is sent to the printer device (using project/print), the printer would eject a page when it encounters this code.

1.75 find/insert code ESC

find/insert code ESC of: FIND MENU

Inserts an ESC code (ASCII 27) at current cursor position. This code introduces many command sequences understood by the printer device respectively the console device (see FIND/ASCII insert).

1.76 find/functions

find/functions of: FIND MENU

QuickFunc

Scans current text for functions, structure definitions or AutoDoc entries. A listview is made up for fast selection: click at a function name to jump to where this function is defined. Several scan modes are available: Select ANSI-C if you are looking for C-functions, PASCAL if you are looking for Pascal procedures/ functions, BASIC if you are looking for Basic PROCEDURES or SUBROUTINES, ASSEMBLER if you are looking for labels beginning with an underscore (e.g. `_main`), C-HEADER if you are looking for structure definitions and finally AUTODOC to scan AutoDoc-like files (programmer manuals). QuickFunc heavily depends on the way of formatting: Don't use a left margin in source codes. Function definitions must start at column one. Some indentation schemes are not handled properly. For example C-function headers (function name, parenthesis, arguments, parenthesis) must fit within one line to be recognized.

You may preselect a default scan mode for special file names using the pattern gadget below the listview. For example you could set the default file extension of the C-mode to `'*.c'`. Thus the C-mode would be used when referencing a file like `'main.c'` (referencing means scanning the file for keywords; see `config/references`).

1.77 CONTROL MENU

CONTROL MENU
menu tree of control menu

control/go to line	control/arrange windows
control/top-bottom	control/center window
control/to last change	control/zip window
control/store position	control/enlarge window
control/recall position	control/freeze window
control/fold all	control/unfreeze window
control/unfold all	control/next window
control/insert-overwrite	control/iconify
control/toggle TAB mode	

1.78 control/go to line

control/go to line of: CONTROL MENU

Asks you for a line to go to. First line of a document is considered to be line 1. Use the 'unfold' (checkmark-)gadget to decide whether you want to have folded sections unfolded if necessary (see Folding). If unfolding is enabled, line numbers are absolute, i.e. if you enter 255, the editor would jump to line 255, no matter whether this line is folded (invisible) so far or not. If you don't enable unfolding, line numbers are based on the number of visible lines. The editor would jump to the 255th visible line. The actual line number of that line - if all folds were unfolded - is either 255 (no folds before this line) or greater (some folds before this line). The line numbers displayed below the window titles are based on the number of visible lines, too. They are not absolute. If you would perform an absolute jump to line 300, a number less than 300 might be displayed if your text contains folds (i.e. if not all lines of the text are visible).

1.79 control/top-bottom

control/top-bottom of: CONTROL MENU

Moves the cursor to line one if it has been positioned near the end of your text so far. Moves it to the last line of your text if it has been close to the top so far.

1.80 control/to last change

control/to last change of: CONTROL MENU

Moves the cursor to the line of last change (or at least close to that position if the last operation was a 'delete line').

1.81 control/store position

control/store position of: CONTROL MENU

Records the position of the cursor & the view area of the text in the window. Use control/recall position to recall the cursor and view area. You may store positions for each text window separately. Since GoldED does support 10 bookmarks for each text you may easily add further menu items similar to this one to remember a second, a third, ..., position (see config/menus for information on modifying menus).

1.82 control/recall position

control/recall position of: CONTROL MENU

Recall cursor position/view stored by control/store position . If you simply want to jump back to the line of your last operation at some time or other you do not have to use bookmarks. Use control/to last change instead.

1.83 control/fold all

control/fold all of: CONTROL MENU

Scans text for fold markers. Folds all text sections surrounded by fold markers (see below for general information on GoldED's folding facilities). There is a keyboard shortcut quite similar to this function: CTRL + HELP; it toggles all folds: if the cursor is placed over a fold header, all folds are unfolded. If the cursor is placed over a normal text line, all sections surrounded by fold markers are folded.

Folding

One of this editor's most useful features is its folding capability: Folding means hiding some lines of a file temporarily. This is quite useful if you are working on a large sourcefile but don't want to get lost in thousands of lines. Simply fold away all functions you are not interested in. Unfold the ones you are working on.

How to fold lines

If you want to hide a section of lines, enclose it into 'fold markers'. Fold markers are just character sequences. Default sequence is `///` (which is regarded as a comment by most K&R compilers respectively all C++ compilers). Example - type:

```
/// "important function"

void
main()
{
    puts("fold me !");
}

///
```

Now place the cursor over any line of the example above (except the last line) and press the HELP key. The lines above will vanish, a single new line - the 'fold header' - appears:

```
> important function
```

To unfold that header, press the HELP key once more. You may fold as many sections of a document as you like. Use CTRL + HELP to unfold all folds upon a single keystroke. You may even have folds within folds up to any level

(nested folding), however this requires the use of different markers for beginning/end of a fold section (see config/misc). Fold markers may be set to any string of up to 8 letters. The shorter the faster. They must start in column 1 to 5 (to speed up scanning). In generally you should choose a sequence regarded as comment by your compiler. Or embed the fold markers into comments - for example after a REM if you are a BASIC programmer.

Fold headers are write protected since they aren't normal lines: keyboard input is suppressed if the cursor is placed over a fold header. Folded sections are not recognized by find & replace operations. Unfold a fold if you want to change it. However, block operations (see block menu) do work. It is possible to copy, duplicate, cut or remove a fold if it has been marked as a block. Saving or printing a file is not influenced by folding - the text is treated as if all folds were unfolded.

1.84 control/unfold all

control/unfold all of: CONTROL MENU

Scans text for fold headers. If some are found, they are unfolded. Have a look at the Folding chapter for information on GoldED's folding facilities.

1.85 control/insert-overwrite

control/insert-overwrite of: CONTROL MENU

Toggles writing mode from insert to overwrite and vice versa. In insert mode keyboard input is inserted into the text without overwriting the existing text. In overwrite mode the existing text is replaced by your input. The current mode is displayed in the window's title bar. It is either INSR (insert) or OVER (overwrite).

1.86 control/toggle TAB mode

control/toggle TAB mode of: CONTROL MENU

Toggles TAB mode from solid to light and vice versa. Solid TAB's actually insert blank characters into your text. Depending on whether insert mode is on or off (see control/insert-overwrite), the existing text is either indented or overwritten. Light TABs do not overwrite anything - the cursor is simply moved to the next TAB position. See config/TABs on how to set TAB positions.

1.87 control/arrange windows

control/arrange windows of: CONTROL MENU

Arranges windows on screen. If you have only one open window, the window is resized to cover the complete screen. If you have two open windows, the windows are made to share the display without overlapping. The screen's title bar is left free. This function does handle autoscroll screens (screens larger than the display) properly - the window(s) are arranged within the visible section.

1.88 control/center window

control/center window of: CONTROL MENU

Centers current window on screen. If the screen is an autoscroll screen (i.e. larger than the display), the window is centered within the screen's visible section.

1.89 control/zip window

control/zip window of: CONTROL MENU

Zips current window as if the window's zip (zoom) gadget were used: The OS keeps track of two alternative window sizes for each window. This functions toggles between them. The minimum window size is limited by GoldED (the editor ensures that at least one line of text can be displayed; besides it keeps the column/line display readable).

1.90 control/enlarge window

control/enlarge window of: CONTROL MENU

Enlarge current window to make it as big as the screen. The screen's title bar won't get covered. This functions handles autoscroll screen (screens larger than the display), too: only the visible section is considered.

1.91 control/freeze window

control/freeze window of: CONTROL MENU

Freeze text buffer, i.e. close the window but don't free the text itself. Useful to have several text files present without cluttering the display. Use

control/unfreeze window from the control menu to reopen the window. Freezing the last window will leave you without menus - use the hotkey to reopen display: right ALT + right SHIFT + RETURN.

1.92 control/unfreeze window

control/unfreeze window of: CONTROL MENU

Asks you for a frozen window to reopen. This functions tries to restore the old window position/size. See control/freeze window for information on how to freeze a window.

1.93 control/next window

control/next window of: CONTROL MENU

Activate the 'next' window. A *very* handy function since you won't have to use the mouse anymore for window activation. Simply flip from one window to another using this command.

1.94 control/iconify

control/iconify of: CONTROL MENU

Iconify GoldED. All windows as well as the GoldED screen (if the editor did use an own screen) are closed. This function depends on the availability of the workbench screen where the editor attempts to place an AppIcon. Iconify won't work if the workbench is closed.

AppIcon

The AppIcon may be used to pass new files to GoldED: Simply drag icons of text files over it. Multi select is supported (i.e. you may drag more than one file at once over the icon using extended selection: hold the SHIFT key down while you select icons). The editor will open a new window for each file. Doubleclick at the icon to wake GoldED up again.

1.95 MISC MENU

MISC MENU
menu tree of misc menu

misc/help	misc/calculator
misc/project setup	misc/new shell
misc/pick line	misc/delete file
misc/push line	misc/HiSpeed
misc/swap lines	misc/insert date
misc/double line	misc/insert time
misc/undo line	misc/command
misc/statistics	misc/last error

1.96 misc/help

misc/help of: MISC MENU

AmigaGuide is made to display the main page of GoldED's manual ('database' in terms of AmigaGuide). You may use config/menus to select a new database. This feature depends on the amigaguide library - if it is not available, you won't get any help.

1.97 misc/project setup

misc/project setup of: MISC MENU

Open project definition requester. Used to specify all source files that belong to a project. Doubleclick on a file being part of that list to load it. GoldED itself does not use this list. The editor only provides a few basic functions to add/remove source files to/from the list. This command has been implemented to support the development of third party make utilities. Further processing has to be done by external programs:

List access (programmers only)

Programs may send a QUERY command to GoldED's ARexx port in order to obtain a pointer to a linked list of nodes: QUERY PRJLIST. Example source code is shipped with GoldED; check the GoldED:Tools/PRJSource directory. The node->ln_Name fields of the list's nodes will point to source file names (NULL terminated). You may pass this list to a listview. Use of this list requires a previous LOCK to ensure a valid list. The list is read-only. Use the PROJECT command to add or remove files. Do not modify the list on your own.

1.98 misc/pick line

misc/pick line of: MISC MENU

Delete current line from text. The line is not lost - it is put to a pick-push ring buffer (last in, first out). You may reinsert it anywhere else

by using `misc/push line` . This function is assigned to the keyboard, too (CTRL + DEL). Example usage: Delete 3 lines, move to another line and call `misc/push line` three times to reinsert the lines. The pick/push buffer holds up to 50 lines.

1.99 misc/push line

`misc/push line` of: MISC MENU

Inserts the last line of the pick/push buffer before current line. See `misc/pick line` for further explanations.

1.100 misc/swap lines

`misc/swap lines` of: MISC MENU

Swaps current line with next line. The function is usually used via the ARexx interface to sort a text.

1.101 misc/double line

`misc/double line` of: MISC MENU

Doubles current line. Faster than using the clipboard or `block/copy` .

1.102 misc/undo line

`misc/undo line` of: MISC MENU

Restores the previous contents of a line after you have changed it. You can undo a line as long as the cursor has not been moved to another line - leaving a line means accepting the change. This function can not be used to undo a delete line operation. Use `misc/push line` to reinsert a deleted line.

1.103 misc/statistics

misc/statistics of: MISC MENU

Presents a statistic overview of a text. Bytes, lines, folded blocks and non-ASCII-characters (codes above 127) are counted. Additionally the width of the longest line is determined. This function treats the text as if it were saved (i.e. all folds unfolded, CR codes appended to the lines).

1.104 misc/calculator

misc/calculator of: MISC MENU

Tries to run the calculator of your workbench (must be placed in sys:tools). The calculator is made to appear on the screen used by GoldED even if a custom screen is used.

1.105 misc/new shell

misc/new shell of: MISC MENU

Opens a shell window on the screen used by the editor. The window is arranged on screen, i.e. it will open within the visible section of overscan screens. GoldED takes care of providing valid path information: if you have a PATH SYS:C2 ADD command in your startup-sequence, the shell window will know about it.

1.106 misc/delete file

misc/delete file of: MISC MENU

Asks you for a file to delete. You are asked for confirmation. It is not possible to delete delete-protected files.

1.107 misc/HiSpeed

misc/HiSpeed of: MISC MENU

Tries to run the HiSpeed printer tool - the program is made to open its window on the screen used by GoldED.

HiSpeed

HiSpeed has been designed for PCL printers like Hewlett Packard's DeskJet family. It is used to reduce the amount of paper when printing large files by

using a small typeface. Up to 8 A4 pages are redirected to a single A4 sheet. It is fast. Speed depends on your DeskJet model; DeskJet+: about 4 pages/minute if you switch to HiSpeed mode. It should be a nice utility for people who have to do a lot of printing. Some of HiSpeed's features are:

- o workbench interface
- o AppWindow/AppIcon support
- o shell interface
- o ARexx port
- o single or double sided printing
- o descending printing available
- o free layout - e.g. two columns
- o ANSI ESC sequences supported
- o Linefeed adjustable
- o spooler (job list)
- o HiSpeed mode: fast printing
- o preview (WYSIWYG)
- o many fonts supported
- o page headers
- o numbering of lines
- o index/appendix creation
- o portrait/landscape
- o book mode to print A5 books

Please read the HiSpeed manual (see GoldED:Tools/HiSpeed) for more information on this tool. HiSpeed is shareware. The registered version is not part of the GoldED light distribution but it is part of the GoldED pro package; see how to register .

1.108 misc/insert date

misc/insert date of: MISC MENU

Inserts current date (e.g. "Boston, 3/5/93") at cursor position. You should set the environment variable USERTOWN to the name of your town using the DOS command setenv (e.g. setenv USERTOWN "Boston"). DOS/setenv does use the 'env:' directory to store environment variables. Unfortunately this directory is placed in RAM, so your settings are lost after a reset. Copy env:usertown to envarc:usertown to prevent this.

1.109 misc/insert time

misc/insert time of: MISC MENU

Inserts time string at cursor position. Formatting depends on the locale library (see language).

1.110 misc/command

misc/command of: MISC MENU

Asks you for one of GoldED' internal commands to be executed (see list of internal commands). This function is assigned to SHIFT ESC, too. Parsing is done by the OS function ReadArgs, which is used by most CLI commands, too, so same rules as usual apply (arguments containing spaces have to be quoted).

1.111 misc/last error

misc/last error of: MISC MENU

Shows the last error message once more (if there has been any so far).

1.112 MACRO MENU

MACRO MENU
menu tree of macro menu

macros/edit macro	macros/config
macros/run text as macro	macros/options
macros/record start-stop	macros/modules
macros/play	macros/compile & link
macros/load	macros/compile & link new
macros/save	macros/display errors
macros/dmake	macros/error: first
macros/compile+link text	macros/error: previous
macros/number	macros/error: next
macros/revision	macros/run executable
GUIMake	

1.113 macros/edit macro

macros/edit macro of: MACRO MENU

Changes to GoldED's macro directory and asks you for an ARexx macro to load. You should save your current text before or open a new window. GoldED's ARexx macros use 'GED' as suffix (e.g. number.ged). All macros addressing GoldED must use a special protocol to register with GoldED before performing operations to prevent race conditions if user & macro are trying to control the editor simultaneously (see ARexx port). I have provided an empty macro (empty.ged) to be used as basis for own development efforts. Simply load this macro, insert your code and save it under a new name.

1.114 macros/run text as macro

macros/run text as macro of: MACRO MENU

Tries to execute the current text as macro. You should save the text before you call this function since the copy on disk is executed, not the text in memory. All ARexx macros have to start with a comment (/* ... */) to get recognized by the ARexx server. If the first line of the text isn't a comment, nothing happens. A script called by this functions has its host set

up properly already (i.e. you don't need an ADDRESS command). Example: type these lines (without left margin) and then select 'run as macro' to have them executed:

```
/* this is a test */  
  
LOCK CURRENT  
REQUEST BODY "Hi!"  
UNLOCK
```

1.115 macros/record start-stop

macros/record start-stop of: MACRO MENU

Macro recording

Start (first call) respectively stop (second call) recording of a keyboard/command sequence. During recording all key presses and menu selections are logged. Mouse movements/clicks are not recorded. Use macros/play to replay a recorded sequence. Use macros/save to write the sequence to disk, ready to be used at some other time. If you open a requester while recording you will be asked whether this requester should pop up in playback mode, too (unless it belongs to the config menu). If you disable the requester, GoldED will use the selections made at recording time (e.g. if you moved the cursor to line 100 during recording, it will be moved to line 100 in playback mode, too. No GOTO requester will appear). Macro recording is based on low-level events like keystrokes or menu selections. For example pressing of the F10 key is recorded as "F10 has been pressed", no matter what action has been assigned to that key. If you change key bindings or menus after recording a macro, the macro will behave differently. This command is assigned to the SHIFT-F10 key, too.

1.116 macros/play

macros/play of: MACRO MENU

Replay a previously recorded keyboard/command sequence (use SHIFT+F10 or macros/record start-stop to enter/leave recording mode). This command is assigned to the F10 key, too. If you want to replay the macro several times, use the misc/command requester instead - for example you could enter MACRO PLAY LOOPS 10 to run the sequence ten times.

1.117 macros/load

macros/load of: MACRO MENU

Load a recorded keyboard/command sequence to be replayed by macros/play (or

by pressing the F10 key). In generally it is more convenient to assign often used sequences to keys or menu items (see MACRO command] instead of using this function.

1.118 macros/save

macros/save of: MACRO MENU

Save a recorded command sequence, created by macros/record start-stop . Sequences related to GoldED should be written to the GoldED:Macros drawer. Default file extension is "*.ged".

1.119 macros/dmake

macros/dmake of MACRO MENU

Does look for a makefile called 'dmakefile' within the directory of the current text. If one is found, dmake of the DICE C compiler is evoked (won't work if you don't own DICE, of course). This menu command actually is a macro. It fails if the ARexx server REXXMAST is not running in the background. Usually the ARexx server is installed during startup (s:startup-sequence): run >NIL: sys:system/REXXMAST.

1.120 macros/compile+link text

macros/compile+link text of: MACRO MENU

Compiles and links the text using DCC of the DICE C distribution (won't work if you don't own DICE). To be more precisely: a temporary copy of your text (T:TEST.c) is created, this is passed to DCC to create an executable called T:TEST. If DCC has compiled/linked your file (without error/warnings), you are asked whether you want to run it. Example: type the following lines (without left margin) and then call the compile & link command:

```
main()
{
    printf("hello world !");
    exit(0);
}
```

1.121 macros/number

macros/number of MACRO MENU

Does number all lines of a text. You are asked for the first line number as well the step value to be used. Choose 1000 respectively 10 if you want to use 1000, 1010, 1020, ... This command actually is a macro. It won't work if the ARexx server REXxMast is not running in the background. Usually the ARexx server is installed during startup (s:startup-sequence or s:user-startup):
run >NIL: sys:system/RExxMast

1.122 macros/revision

macros/revision of MACRO MENU

Revision control (programmers only)

Update version string, revise history: This function will scan your text for a version string according to the programmers' style guide (published by Commodore). If one is found, the revision number is increased by one. Versions strings (e.g. \$VER GED 1.0 (1.10.93)') are emedded into programs to provide required information for the AmigaDos command VERSION (VERSION does scan files for the '\$VER:' keyword). Additionally, you are asked for a short comment about the latest version if a '\$HISTORY:' section has been detected within current text. The comment is added at the top of the history list. Example text header understood by the revision command:

```
char *Version = "$VER: revision 0.8 (21 Jul 1993)";

/*
    $HISTORY:

    18 Jul 1993 : 000.003 : added commandline args
    18 Jul 1993 : 000.001 : initial release
*/
```

This command actually is a macro (done by Marius Gröger). It won't work if the ARexx server REXxMast is not running in the background. Usually the ARexx server is installed during startup (s:startup-sequence or s:user-startup):
run >NIL: sys:system/RExxMast

1.123 macros/GUIMake

macros/GUIMake of: MACRO MENU

GUIMake

Display GUIMake copyright information. GUIMake is a project manager for DICE-C. The GUIMake package has been developed by Rico Krasowski - included with kind permission of the author. The purpose of GUIMake is to replace makefiles by a GUI-driven program. Besides comfortable file management

(compiling/linking) GUIMake offers very handy error handling facilities. For example GUIMake might make the editor jump to the first erroneous line after compilation. GUIMake is a stand alone program -communication between GUIMake and GoldED is based on ARexx. Have a look at the GUI guide file (tools directory) for more detailed information on GUIMake and its ARexx port. This function as well as all other menu entries related to GUIMake are not available if you have disabled DICE support during installation of GoldED. GUIMake access depends on the ARexx server REXXMast which must be running in the background. Usually the ARexx server is installed during startup (s:startup-sequence): run >NIL: sys:system/REXXMast.

1.124 macros/config

macros/config of MACRO MENU

Open config window of GUIMake (used to set up a project). Have a look at the GUIMake manual for a more detailed description. This function isn't available if you have disabled DICE support during installation of GoldED.

1.125 macros/options

macros/options of: MACRO MENU

Open options window of GUIMake (used to set compiler options). Have a look at the GUIMake manual for a more detailed description. This function isn't available if you have disabled DICE support during installation of GoldED.

1.126 macros/modules

macros/modules of: MACRO MENU

Open modules window of GUIMake. Used to select files related to your project: source files, header files, ... Have a look at the GUIMake manual for a more detailed description of Rico's make utility.

1.127 macros/compile & link

macros/compile & link of: MACRO MENU

Compile & link files of your current project. See macros/GUIMake for information on how to set up a project. Won't recompile a file unless it has been changed since last compilation (see [macros/recompile & link]). Have a look at the GUIMake manual (BUILD command) for a more detailed description.

This function isn't available if you have disabled DICE support during installation of GoldED.

1.128 macros/compile & link new

macros/compile & link new of: MACRO MENU

Compile and link all files of your current project, no matter whether files have been changed or not since last compilation. Use macros/GUIMake to set up a project (i.e. to specify source files, header-files and object files related to your project or to set compiler options). Have a look at the GUIMake manual for a more detailed description. This function isn't available if you have disabled DICE support during installation of GoldED.

1.129 macros/display errors

macros/display errors of: MACRO MENU

Load error file created during compilation (see macros/compile & link). This function isn't available if you have disabled DICE support during installation of GoldED.

1.130 macros/error: first

macros/error: first of: MACRO MENU

Make GoldED jump to the first erroneous line (either error or warning) after compilation (see macros/compile & link) has completed. The original DICE error/warning text is displayed below the window's title bar. This function isn't available if you have disabled DICE support during installation of GoldED.

1.131 macros/error: previous

macros/error: previous of: MACRO MENU

Make GoldED jump to the 'previous' error/warning line (after having used macros/error: next). The original DICE error/warning text is displayed below the window's title bar. This function isn't available if you have disabled DICE support during installation of GoldED.

1.132 macros/error: next

macros/error: next of: MACRO MENU

Make GoldED jump to the 'next' error/warning. The line number is extracted from the file T:ERRORS produced by the compiler/GUIMake during compilation (see macros/compile & link). The original DICE error/warning text is displayed below the window's title bar. This function isn't available if you have disabled DICE support during installation of GoldED.

1.133 macros/run executable

macros/run executable of: MACRO MENU

Run the executable created by DICE/GUIMake (see macros/compile & link). This function isn't available if you have disabled DICE support during installation of GoldED.

1.134 CONFIG MENU

CONFIG MENU
menu tree of config menu

config/keyboard bindings	config/menus
config/TABs	config/printer
config/references	config/misc
config/dictionary	config/calender-time
config/indentation	config/save
config/display	config/load

Use these functions to adjust GoldED to your likings; don't forget to use config/save before you leave the editor; your definitions would be lost otherwise.

Configuration guidelines

I'd like to have other people developing their own menus or keyboard assignments and release them to the public. I'd like to include some into the unregistered release of GoldED, too. Useful guidelines for contributions:

Use standard shortcuts for standard menu functions. For example, do not decide for A as shortcut for 'open file' if all other editors use O.

Provide a guide help file (AmigaGuide) for your menu definitions respectively a documentation of your keyboard assignments. Use the same language for menus & help text. Feel free to copy parts of the original GoldED manual(s).

Use a symbolic path when referencing external files/programs through menu calls. Suggested: 'GoldED:tools' for DOS tools respectively 'GoldED:ARexx' for ARexx scripts. Do not use something like this: 'dh0:c/muchmore'.

Provide an 'Installer' script to set up a working environment. Feel free to copy parts of the original script(s) that came with GoldED.

1.135 config/keyboard bindings

config/keyboard bindings of: CONFIG MENU

Open keyboard binding requester used to map desired action to keys or key-qualifier combinations (GoldED does recognize the qualifiers ALT, CTRL and SHIFT). As long as you don't bind any action to a key, the default keymap is used whenever it is possible: Pressing the 'A' key would insert an 'A' at current cursor position. Non-character keys (like the cursor keys) are initially unset, i.e. they wouldn't have the expected effect like moving the cursor (however GoldED is shipped with this stuff already set up). To make the cursor keys move the cursor you would have to bind 'move cursor' commands to these keys. Setting up a key(-combination) is easy: Simply use the record gadget and press the desired keys. A new requester will pop up, asking you for what action is to be assigned to this event (see Event definition).

1.136 Event definition

Event definition

You may assign any of the editor's internal commands to a given event (event = keystroke or menu selection). Or a DOS command. Or a macro. Or just a string. Write the command/script name respectively string to the CMD gadget and set the event type; supported event types are internal, arexx, shell, text or dummy. If you set the event type to 'dummy', you would disable the event (i.e. nothing would happen if the event is encountered). The DIR gadget may be used to set the current directory while the event is processed. This is supported for events of type shell only. You may set the output handle (e.g. 'con:0/0/640/400') using the OUTPUT gadget; this is supported for events of type 'arexx' and 'shell' only. GoldED defaults to opening a console window on its screen if the output handle is omitted. The string contents of the CMD/OUTPUT/DIR gadgets are 'interpreted' before use (except if the event is of type 'internal'); see Magic codes.

Select 'shanghai' if you want to make all windows use GoldED's screen during event processing. You have to specify how long shanghai mode is to be turned on; units are seconds. Toggle the AutoSave gadget to ON if you want the text to be saved before the event is processed. Turn Async ON to make events of type 'shell' asynchronous. Finally you may assign a help text to an event using the 'hyper' gadget: enter a node name of the 'current' AmigaGuide database (see config/menus). The help text assigned to a menu event is displayed during MenuHelp processing. If you don't provide a node name, GoldED defaults to use M<menu number>.<item number> (e.g. M1.2). You may use

the '@' character to select a database (i.e. to override the default database) . Example usage: GOLDED:CMD.GUIDE@MAIN would make the editor look for a node 'MAIN' within the amigaguide file 'GOLDED:CMD.GUIDE'.

Multiple commands

Assignment of multiple commands to events is not supported directly. However you may (besides using an ARexx script) set up menu items of type ARexx to be send to the server directly. The ARexx server does support the separator ";". Example to open the info requester twice (might be mapped to a menu or a key):

```
'INFO USER; INFO USER'
```

Commands of type 'ARexx' are processed asynchronously (i.e. they are sent to the server without blocking GoldED). This may cause problems during playback of recorded sequences (see [macros/playback]) since GoldED won't be able to wait for completion of ARexx messages passed to the server: The sequence might behave differently at playback time compared to recording time.

1.137 Magic codes

Magic codes

Interpretation of strings (see event definition) means that some predefined symbols like \DATE are replaced by their actual value if the string is finally referenced by GoldED; quote a string if you don't want it to be interpreted. The following keywords are supported:

```
"<characters>" ..... character constant
%<number> ..... inserts ASCII code <number>
\n ..... return
\t ..... tab
\b ..... move cursor left
\" ..... quote
\NAME ..... current file name
\CON ..... window dimension string
\DATE ..... current date
\TIME ..... current time
\SCREEN ..... screen name
\HOST ..... name of GoldED's ARexx port
$<name> ..... environment variable <name>
```

1.138 config/TABs

config/TABs of: CONFIG MENU

Open TAB configuration requester. GoldED does support several modes as far as TABs are concerned: solid TABs as well as light TABs are available (see:

control/toggle TAB mode . Additionally you may decide for distinctive TABs, regular ones or dynamic TABs: while regular TABs are set using the 'tab key' slider (e.g. to every 4th column), distinctive ("fixed") TABs are set using a listview; simply enter desired TAB positions. Last but not least dynamic TABs are available: In dynamic mode GoldED will examine the last line(s) to determine appropriate TAB positions. If none are found, regular TAB's are used.

GoldED does never use TABs when writing a file to disk, however some other editors do, usually replacing eight spaces by a single TAB. GoldED does resubstitute these TAB codes by spaces while loading (see project/load). Usually one TAB code is replaced by eight spaces. Use the 'tab file' gadget to change this. Setting it to four would make GoldED use four spaces for each TAB.

1.139 config/references

config/references of: CONFIG MENU

Open requester to set up the editor's reference system. Whenever the user asks for a reference (see find/reference), GoldED does examine its reference database. This database consists of keyword-filename pairs: one reference file for each keyword (e.g. RastPort) GoldED knows about. If a match for the user's request is found, the corresponding reference file is loaded.

Database creation

Use the GET gadget respectively the string gadget to select a database; if you select a non-existent one, it is created (the editor is shipped without a database; suggested file name is 's:GoldED.refs'). A new database initially is empty. All you have to do in order to fill the database is selecting some files using the ADD gadget or directories using the ADD DIR gadget: GoldED will scan all selected files for keywords. Scanning mode depends on the file name: if the file name suffix is '.c', function names are extracted. If the suffix is '.h', structure definitions are extracted. Several other types are recognized, too (see find/functions). You may even change the default file extensions recognized by the scanner (see find/functions).

1.140 config/dictionary

config/dictionary of: CONFIG MENU

Opens requester to edit the dictionary. The dictionary is used by the editors APC facility (to complete expressions) as well as by its AutoCase support:

AutoCase

If you enable AutoCase checking, the editor will search the current line for words present within the dictionary, too. This check is performed when the

cursor leaves the current line. It is case-insensitive (e.g. Rastport and RastPort would be recognized as the same expression). If a match is found, it is replaced by the dictionary entry, thus possibly correcting case.

General hints

Don't make the dictionary too big - the smaller it is, the more efficient it will be. Only add unique names to it - otherwise the editor might try to correct the spelling of a word even if you don't want it. For example it wouldn't be a good idea to add 'RastPort' for this would prevent you from using a variable 'rastport' within your program. However, 'struct RastPort' is fine, since this is the only way to write this kind of structure definition (at least as far as Amiga C programmers are concerned).

Parenthesis check

Toggle the ()-check gadget to ON if you want the current line to be checked for correct use of braces as soon as the cursor leaves it (see find/check). Unfortunately this check is performed if the display is shifted, too, due to the internal design of GoldED. Don't use this option for 'free-style' programming languages like C.

1.141 config/indentation

config/indentation of: CONFIG MENU

AutoIndentation , SmartIndentation

Open requester to set the indentation scheme. Turn AutoIndentation ON, if you want the cursor to be indented the same amount as the previous line after a CR (see return key). Turn SmartIndentation ON if you want automatic indentation after user-defined keywords (e.g. after IF); use the listview gadget to set desired keywords.

1.142 config/display

config/display of: CONFIG MENU

Display mode Font

Open requester to set up display (i.e. resolution, fonts, window look and more). Some gadgets of this requester are 'dangerous' since they require closing down the current windows/display temporarily: the editor might not be able to reopen its display if you are very short of memory.

You may select three different fonts to be used for the text, within requesters or for menus (if GoldED runs on a custom screen). These fonts (apart from the screen font) have to be fixed width fonts.

3D-Look

The 3D gadget toggles the OS2 color scheme (new look) on/off. Choose the mode you like better; the old look is a good choice if the background color is black. GoldED won't use icons within its requesters if the new look is disabled.

Borderless

Two basic window types are available: Borderless windows (looking like a screen) and normal windows. Borderless windows are disabled if GoldED doesn't run on a custom screen.

Shanghai

Turn shanghai mode ON if you want to force all windows usually opening on the workbench screen to open on GoldED's (custom) screen. Use not recommended if GoldED's screen is a one-plane screen (many programs won't look that pleasing then).

1.143 config/menus

config/menus of: CONFIG MENU

Open menu requester; used to set up menus. You may have as many menu titles (left listview) as you like - up to intuition's limit. Each menu may have as many menu entries as you like (right listview) - up to intuition's limit once more. Doubleclick at a menu item to set its action (see event definition). Use the '^' gadget to move a listview entry one position up (respectively down if you use the SHIFT key simultaneously). The 'hyper' string gadget selects a guide file ('database'), containing menu descriptions. This database is used as default database by all commands related to AmigaGuide handling (e.g. misc/help).

1.144 config/printer

config/printer of: CONFIG MENU

Printer definition requester. Used to define printing mode for project/print as well as block/print. Since all output of GoldED goes to the standard printer driver, this works with any printer (while misc/HiSpeed does support PCL printers only). Contents of the 'init' string gadget are sent to the printer after all other initialization (e.g. quality selection) has been done. This gadget may be used to pass printer specific data, too: The gadget's contents are interpreted (see magic codes), you may easily pass a so called aRaw to the printer device: '27 [<bytes> 34 r'. To send a 7-bytes command (e.g. "0123456") to the printer, enter:

```
%27 "[7" %34 "r" "0123456"
```

1.145 config/misc

config/misc of: CONFIG MENU

Open main preferences requester. This requester is used to set several 'Auto' features as well as the backup handling, default protection bits, fold markers and XPK compression mode (see project/save as XPK):

AutoFold AutoLoad AutoCenter

If AutoFold is enabled, the editor does look for fold markers after loading; if some are found, the corresponding sections are folded (see Folding); happens before the text is displayed. Use the fold marker gadgets (start/end) to set the marker sequence for start respectively end of folds. The shorter this sequences are, the faster folding will be. Using the same sequence for marking start/end of sections to fold speeds up folding even more. However, this would prevent you from using nested folding: nested folding requires different markers.

If AutoLoad is enabled, GoldED attempts to load your last project during startup - unless file names are specified.

If AutoCenter is enabled, window positions are not read from the configuration file. Instead, GoldED attempts to center windows on screen (only visible section considered).

XPK compression mode

The XPK listview presents a list of available XPK compressors, found within the libs:compressors subdirectory. Additionally you may enter a password (used by some XPK encryption libraries) and set efficiency for compression (0% to 100%). Please read the original XPK documentation.

Backup creation: AutoBackup

The editor attempts to backup old copies of a file before saving a new version if 'create *.bak' is enabled (otherwise the old version is overwritten). Backups are written to any path you like (see backup path gadget). Enable AutoSave if you want backups of your windows every x minutes (use gadget below AutoSave to enter period). Toggle 'ask' gadget to ON, if you want to be asked for confirmation of AutoBackup events.

Create *.info

If 'create *.info' is ON, the editor does generate an icon for each file saved to disk (unless an old one exists). The default tool of icons created by the editor is GoldED.

File protection bits

These gadgets (read/del/write/script) are used to set the default bits for a new text (e.g. created by project/more ed). Use project/bits to set the actual bits of a document.

Startup macro

The startup ARexx macro is executed once (asynchronously) after the editor has been launched, windows already open. Have a look at the section about the editor's ARexx port as far as ARexx programming is concerned. You could use the macro to customize the GoldED environment according to the type of files loaded during startup (e.g. switch to a C programming environment). Leave the startup gadget empty if you don't need automatic execution of a startup macro.

HotKey support (stay-in-ram gadget)

GoldED does support HotKey activation: if hotkey activation is enabled (stay-in-ram gadget checked) the editor will not be removed from memory even after the last window has been closed. Instead it will wait for a hotkey combination (right SHIFT & right ALT & RETURN) before it attempts to reopen its screen again. HotKey activation will give you a quick response time though it will consume some memory, too. You might want to disable this feature if you are short of RAM. Use the QUIT UNLOAD command (see misc/command) to remove the editor completely from RAM (you could use the commodities exchange program of your workbench, too).

1.146 config/calender-time

config/calender-time of: CONFIG MENU

Attempts to open the time/date preferences program of your workbench (must be placed within sys:prefs).

1.147 config/save

config/save of: CONFIG MENU

Save configuration to a preferences file. Has to be used after GoldED's settings have been changed if you don't want to lose your definitions. Default settings file is envarc:GoldEd/GoldED (this file is used during the editor's startup if no other configuration is specified).

1.148 config/load

config/load of: CONFIG MENU

Load a configuration file & adjust to the new settings. This operation includes closing down and reopening all windows; might be a dangerous call if not enough memory is available (i.e. if the editor isn't able to reopen the windows).

1.149 Keyboard

Keyboard

Please read this chapter carefully if you want to take full advantage of this editor's features. Keys usually perform different tasks depending on what qualifier key(s) is/are pressed simultaneously. Qualifier keys are SHIFT, ALT or CTRL. For example the cursor keys map to seven different functions. Key bindings are not fixed; use config/keyboard bindings to adjust them to your likings. This manual does describe the default setup. The following descriptions are available:

Cursor keys	TAB key
HELP key	ESC key
RETURN key	F-keys
DEL key	

1.150 Cursor keys

Cursor keys Speeds of scrolling

UP/DOWN + ALT

This sequence provides fast scrolling (up or down) - it is one of GoldED's most useful key combinations. The cursor won't move during fast scrolling (i.e. it will stay in the middle of the screen if it was there before you switched to fast scrolling).

UP/DOWN + SHIFT

Go to next (DOWN) respectively previous (UP) page. Pages do overlap a bit to make navigation more comfortable.

UP/DOWN + CTRL

Fast jump: the cursor moves to the next quarter of your text. Useful to roughly set a new position before using fast/normal scrolling for fine tuning.

LEFT/RIGHT + ALT

Shifts the display area to the left respectively right. Usually display is shifted automatically if the cursor reaches the right/left window borders. This function is useful if you want to shift the display without moving the cursor at all.

LEFT/RIGHT + CTRL

This is a shifting function: mark some lines using block/mark, then use this key combination to shift (indent) the block. Mainly used by programmers to ensure a proper indentation scheme. Usually the marked lines are shifted by one column; use the SHIFT key simultaneously to set shifting distance to TAB

distance.

LEFT/RIGHT + SHIFT

Moves cursor to the beginning of the next (RIGHT) respectively previous (LEFT) word.

LEFT/RIGHT + SHIFT + ALT

Moves cursor to the end of the current/next (RIGHT) respectively previous (LEFT) word.

1.151 HELP key

HELP key

HELP

Fold/unfold current section: Unfold if cursor is placed over a fold header (see Folding), otherwise look for fold markers & fold lines between markers (cursor must be placed between a fold start marker and a fold end marker).

HELP + CTRL

Fold/unfold the whole text: Unfold all folded sections if cursor is placed on a fold header (see Folding), otherwise fold all sections surrounded by fold markers.

1.152 TAB key

TAB key

TAB (+ SHIFT)

Move cursor to next TAB position. This editor does support simple/regular TABs (e.g. every 4th column) as well as fixed TABs (any column you want) and dynamic TAB's. Use config/tabs to set mode & TAB positions. Press the SHIFT key simultaneously if you want to jump to the previous instead of the next tab position (backtab). TABs are either solid (i.e. they behave as if they were a sequence of spaces) or light (i.e. they simply move the cursor without inserting any character); use either control/toggle tab mode or config/tabs to switch from light to solid and vice versa.

Dynamic TABs

GoldED does support dynamic TAB's (suggested by David Gröhler): in dynamic TAB mode (set by config/TABS the editor will examine the previous line(s) to determine appropriate TAB positions. Quite useful if you are about to edit assembler sources. Dynamic TABs default to regular TABs, if the lines above the current line are empty.

TAB + ALT (+ SHIFT)

Usually either distinctive or normal TABs are active. However, you can switch to distinctive tabs on the fly by holding down the ALT key while using TAB or TAB SHIFT.

1.153 RETURN key

RETURN key

RETURN

Split current line at cursor position & move cursor to next line. This editor does support AutoIndention : if you press the RETURN key, the current line's indention is used as default indention for the next line (i.e. if the current line is indented by four columns, pressing <CR> will move the cursor to the fourth column of the next line).

RETURN + SHIFT

Same as RETURN but the current line is not splitted no matter where the cursor is positioned so far.

RETURN + CTRL

Inserts an empty line: the cursor is not moved at all but a new line is inserted before the current line.

1.154 DEL key

DEL key

DELETE (+ SHIFT)

Delete character at current cursor position, shift rest of line one position to the left. Press the shift key simultaneously if you want to delete until the end of line. You can undo this operation using misc/undo line .

DELETE + CTRL

Delete the current line. You can recall up to 50 deleted lines using misc/push line since lines are not actually lost but put to the pick/push buffer.

DELETE + ALT

Delete the next word. You can undo this operation as long as the cursor doesn't leave the current line using misc/undo line .

1.155 ESC key

ESC key

Trys to 'complete' the word your cursor is placed over. Example usage: type 'TIG', then press the ESC key. 'TIG' would be replaced by 'TAG_IGNORE' if the C-dictionary is present (see config/dictionary on how to load/edit/create a dictionary). GoldED does use a (simple) pattern matching algorithm to find appropriate dictionary entries, so you might use other abbreviations than 'TIG', too (e.g. 'TAGI'). This will work as long as the first letter of the short form is the first letter of the full form. However, the larger your dictionary grows the more detailed your abbreviations have to be to ensure unique identification.

1.156 F-Keys

F-Keys

Use the config/keyboard bindings menu to assign strings, shell commands, ARexx macros or one of the editor's internal commands to any key including the function keys. However, some commodities do their own function key mapping. If such a commodity is installed, the editor won't notice function key events. Default mappings of the function keys are:

key	decription	see

F1	open file	project/load
F2	merge file	project/insert
F3	print file	project/print
F4	hide block	block/hide mark
F5	mark line	block/mark
F6	find next	find/find next
F7	next page	cursor keys
F8	set insert mode	control/insert-overwrite
F9	toggle TAB mode	control/toggle TAB mode
F10	play macro	macros/play
SHIFT		
& key	decription	see

F1	save as	project/save as
F2	save as XPK	project/save as XPK
F3	quit	project/quit
F4	clear text	project/clear text
F5	mark line	block/mark
F6	replace	find/replace
F7	page up	cursor keys
F8	overwrite mode	control/insert-overwrite
F9	project setup	misc/project setup
F10	play macro	macros/record start-stop

1.157 ARexx port

ARexx port

ARexx macros vs. recorded sequences

GoldED does offer two kinds of macros for automated control: ARexx scripts and recorded sequences. ARexx scripts are programs quite similar to programming languages like BASIC. They are evoked by GoldED (e.g. by setting up a menu item of type 'ARexx'; see config/menus), but actually executed by the ARexx master server (part of the Amiga operating system). The ARexx server will examine the script and send messages to GoldED respectively receive messages from GoldED during execution. For example the ARexx server could ask GoldED to jump to a special line if it detects a GOTO command within a script. GoldED would tell the ARexx server whether the operation has been successful. Due to the flexibility of ARexx this is a very powerful method to automate control of GoldED. However this approach requires at least some basic knowledge of ARexx. If you need automated control but are not interested in ARexx you might want to use GoldED's ability to record command sequences instead: Enable recording using macros/record start-stop and perform a sequence of commands. In other words: make the editor 'learn' how to do it. Once you have recorded a sequence you may replay it as many times as you like (see macros/play). You may save sequences macros/save) or assign them to events like menus or keystrokes using GoldED's MACRO command. Sequences are far less powerful than ARexx script. But they are far more handy, too.

This section does describe the editor's ARexx interface. You are expected to be familiar with ARexx basics, i.e. you should know about the purpose of ARexx, how to write scripts, how to talk to applications, ...

ARexx basics

ARexx control of this editor is somewhat complicated since you never know how many editor tasks are running, how many windows are open or what the user is doing when ARexx wants to take over control. It is therefore quite important to obey to certain rules which are to be discussed now:

1. Select a host
2. Lock a window
3. Do your job
4. unlock GUI

1.158 Select a host

Select a host

If you run ARexx scripts from within the editor (i.e. if you set up menu items of type 'ARexx' or if you execute the current text as macro using macros/run text as macro), any script commands which are not part of ARexx itself are sent to the editor (the 'host') automatically. However, if your script is evoked from a different program (e.g. rx), it will have to select a port for communication: use ADDRESS <port name> for this purpose.

Port name

The editor's ARexx port is called "GOLDED.1" if the editor is run only once. The ARexx port of a further editor task would be "GOLDED.2". Select project/about if you want to know the current port/screen name. Or use the QUERY command (with the HOST argument) from within a script. As long as you use the QuickStarter to run GoldED, you usually won't have to deal with port names different from 'GOLDED.1'.

1.159 Lock a window

Lock a window

Your script has to tell the editor what window is going to be affected - use the LOCK command (e.g. 'LOCK CURRENT' to lock the current window). After locking a window, the GUI is locked, too, to prevent the user from disturbing the macro. If you don't use the lock command, your macro might still appear to work perfectly, but it will break under special circumstances (e.g. if the user closes a window while a script is executed). Once you have locked a window successfully (ARexx return code RC is 0), you have to take care of unlocking it again on termination of your script (see Unlock GUI).

1.160 Do your Job

Do your Job

You can use any of the editor's internal commands within your macro. Please keep in mind that commands sent from ARexx to GoldED are parsed twice: first by the ARexx server while executing the script, second by GoldED using the ReadArgs() function of the dos library. This sometimes screws things up a bit - especially as far as quotes are concerned. Suggestion: Write the lines of your script as if you were talking directly to GoldED: quote strings, command names uppercase (step 1). Then put the lines to be sent to GoldED into single quotation marks to mark them as commands (step 2). Finally double single quotation marks within these lines to prevent ARexx from regarding them as string delimiters (step 3). Example:

```
step 1: REQUEST BODY "Hi, I'm an empty macro"
step 2: 'REQUEST BODY "Hi, I'm an empty macro"'
step 3: 'REQUEST BODY "Hi, I''m an empty macro"'
```

Usually GoldED does pass command results to your script using the special ARexx variable RESULT - at least if you have asked for results using OPTIONS RESULTS. Some commands like QUERY support specification of a variable name, too, using the VAR/K option. Example: 'QUERY ABSLINE VAR LINE'

No result is returned if a command fails - instead the special variable RC is set to the error level: 5 = warning, 10 = error, 20 = fatal error. RC would be 0 if a command has been successful. You have to use the OPTION FAILAT command of ARexx to prevent ARexx from stopping execution if RC is not NULL,

i.e. to receive RC return codes at all. The special variable RC2 will keep an error text if a command has failed (i.e. if RC is not 0).

1.161 Unlock GUI

Unlock GUI

An ARexx script must unlock the GUI before it terminates, if a prior call to Lock (see Lock a window) has been successful. It mustn't use unlock if a prior attempt to lock has failed. Omitting unlock will leave the editor dead-locked, so take care to unlock the GUI even if your script breaks (maybe due to a syntax error). This can be achieved using the error handling facilities of ARexx (e.g. SIGNAL or OPTION FAILAT). Just have a look at the scripts in the GoldED:ARexx drawer. As a last resort the Unlock macro is provided: simply doubleclick at its icon - all editors will be unlocked (which is a dangerous operation if one of those tasks is processing an ARexx script).

1.162 Internal commands

Internal commands

GoldED offers a set of about 250 commands (see Command list), supported by all interfaces of GoldED: you may use them in ARexx macros, bind them to menu items (see config/menus), attach them to keys (config/keyboard bindings) or execute them directly using misc/command . It is possible to combine several functions (see multiple commands). As far as arguments are concerned, the DOS rules apply since GoldED does use the ReadArgs function of OS2.0 just like most CLI commands: strings containing spaces must be quoted, option and keywords may be uppercase or lowercase. Command templates/options are described in the same way as CLI commands are described. Example:

```
PRINT FORCE/S,ITALICS/K,ALL/S,LPI/N,CONFIG/K
```

This PRINT command obviously accepts five options: force, italics, all, lpi and config. The option force is a switch (/S): it does make the print command behave in a special way described in this manual _if_ this option is specified. The second option introduces a keyword (/K) - value pair; e.g. print italics=true. The equality sign may be omitted. Due to ReadArgs() parsing, you will have to use *" instead of " when embedding quotation marks into strings to prevent the parser from considering a quote as start/end of a string (** results in a single *). Options of type '/K' - just like any other options apart from /A ones - do not have to be specified. If they are specified, a further argument (like true) is expected. Supported arguments depend on the command: If one of the commands below offers an option described as BOOL, it would accept the strings true, false and probably toggle, too. If a command's option is marked as STRING, any text string is accepted as argument (e.g. print config "S:prt.prefs"). The 4th keyword in the example above (lpi) introduces a numerical (/N) argument; example: print

lpi=10. The equality sign may be omitted once more. The valid argument range depends on the command (e.g. byte, word, unsigned word or long). Further option types are "\N" (accepts rest of line as string), "\M" (accepts multiple strings) and "\A" (means: this argument must be specified).

1.163 Command list

Command list (use: see internal commands):

BACK	DOWN	HELP	MODE	PROJECT	TABS
BIND	DPAGE	INDENT	MORE	PUSH	TASK
BITS	ENDWORD	INFO	NAME	QUERY	TEXT
BLOCK	EVENT	INSERT	NEW	QUIT	UJUMP
BRACKET	EXALL	KEY	NEXT	REFRESH	UNDO
CLIP	FDOWN	LAYOUT	OPEN	REMAP	UNLOCK
CMD	FILE	LEFT	PATH	REPLACE	UP
CODE	FIND	LINES	PHRASE	REQUEST	UPAGE
COLON	FIRST	LOCK	PING	RIGHT	USE
CONFIG	FOLD	MACRO	PONG	RX	VIEW
CR	FORMAT	MARK	POP	SAVE	WINDOW
DEL	FREEZE	MAXDOWN	PREFS	SCREEN	XREF
DELETE	FUNC	MAXUP	PREV	SHIFT	
DIR	FUP	MENUS	PREVEND	SMARTCR	
DJUMP	GOTO	MISC	PRINT	TAB	

1.164 BACK

BACK

Description of internal commands

command	option	description

BACK	(no options)	backspace operation

1.165 BIND

BIND

Description of internal commands

command	option	description

BIND	(no options)	open keyboard requester (command assignement)

1.166 BITS

BITS

Description of internal commands

command	option	description

BITS	ASK/S	open a requester to edit protection bits/comment
	R/K	set readable bit (BOOL)
	W/K	set writeable bit (BOOL)
	D/K	set deletable bit (BOOL)
	S/K	set script bit (BOOL)
	COMMENT/K	set comment (STRING)

1.167 BLOCK

BLOCK

Description of internal commands

command	option	description

BLOCK	UPPER/S	make block uppercase
	LOWER/S	make block lowercase
	SORT/S	sort block
	COPY/S	copy block to cursor position
	MOVE/S	move block to cursor position
	HIDE/S	hide marker after operation

1.168 BRACKET

BRACKET

Description of internal commands

command	option	description

BRACKET	CHECK/S	check use of () in current line
	MATCH/S	move cursor to matching bracket

1.169 CLIP

CLIP

Description of internal commands

command	option	description
---------	--------	-------------

CLIP	CUT/S	move block to clipboard
	COPY/S	copy block to clipboard
	PASTE/S	insert clipboard contents at cursor position

1.170 CMD

CMD

Description of internal commands

command	option	description
CMD	(no options)	open command requester

1.171 CODE

CODE

Description of internal commands

command	option	description
CODE	SHOW/S	show ASCII code of character under cursor
	SET/N	insert code (UBYTE)
	ASK/S	ask for ASCII code to be inserted
	TABLE/S	open character set table requester
	TOGGLE/S	change case of character under cursor

Comment: The SET option is influenced by current writing mode: in insert mode a character is inserted, in overwrite mode the character under the cursor is overwritten.

1.172 COLON

COLON

Description of internal commands

command	option	description
COLON	(no options)	insert semicolon and possibly a CR (return)

Comment: Suggested use is mapping to the ;-Key. Useful for C/C++ programmers. The editor tries to figure out whether a CR should be inserted (e.g. no CR is inserted if the semicolon is part of a 'for' statement). Press CTRL simultaneously to disable CR insertion temporarily.

1.173 CONFIG

CONFIG

Description of internal commands

command	option	description

CONFIG	SPLIT/K	output path prefix (STRING)

Comment: This function splits the current configuration into several files (menu definition file, dictionary, ...). Mainly used during development of GoldED.

1.174 CR

CR

Description of internal commands

command	option	description

CR	(no options)	<RETURN> command; splits line at cursor position

Comment: This function is influenced by the current setup (e.g. by the indentation mode settings).

1.175 DEL

DEL

Description of internal commands

command	option	description

DEL	(no options)	deletes character under cursor

1.176 DELETE

DELETE

Description of internal commands

command	option	description

DELETE	WORD/S	delete next word
	EOL/S	delete until end of line
	LINE/S	delete current line

BLOCK/S	delete block
COLUMN/S	delete column (see AT/N) from block
AT/N	column to be deleted (UWORD); defaults to current

1.177 DIR

DIR

Description of internal commands

command	option	description
DIR	ASK/S	open requester to set 'current directoy'
	NEW/F	set 'current directory' (STRING)

Comment: the current directory is passed to any program run by GoldED (e.g. a shell using misc/new shell). It is used by many of GoldED's internal functions, too.

1.178 DJUMP

DJUMP

Description of internal commands

command	option	description
DJUMP	(no options)	jump to end of screen / next page

Comment: Cursor jumps to last line of screen if placed above that line so far. Jumps to next page otherwise. Compare: DPAGE .

1.179 DOWN

DOWN

Description of internal commands

command	option	description
DOWN	(no options)	move cursor one line down

1.180 DPAGE

DPAGE

Description of internal commands

command	option	description

DPAGE	(no options)	show next page (compare: DJUMP)

1.181 ENDWORD

ENDWORD

Description of internal commands

command	option	description

ENDWORD	(no options)	move cursor to end of word

1.182 EVENT

EVENT

Description of internal commands

command	option	description

EVENT	RAW/S	under development - don't use
	KEY/N	under development - don't use
	SHIFT/S	under development - don't use
	ALT/S	under development - don't use
	CTRL/S	under development - don't use
	NDEF/S	under development - don't use
	INTERN/K	under development - don't use
	SHELL/K	under development - don't use
	AREXX/K	under development - don't use
	TEXT/K	under development - don't use
	PATH/K	under development - don't use
	OUTPUT/K	under development - don't use
	HYPER/K	under development - don't use
	ASync/K	under development - don't use
	ASAVE/K	under development - don't use
	SHANGHAI/N	under development - don't use

1.183 EXALL

EXALL

Description of internal commands

command	option	description
EXALL	(no options)	Examine text

Comment: To be used within ARExx macros only. Used to update variables related to text statistics (see QUERY).

1.184 FDOWN

FDOWN

Description of internal commands

command	option	description
FDOWN	(no options)	scroll down in fast mode

1.185 FILE

FILE

Description of internal commands

command	option	description
FILE	DELETE/S	delete file
	NAME/K	file to delete (STRING)
	FORCE/S	don't ask for confirmation

Comment: Protected files are not deleted unless the FORCE mode is used.

1.186 FIND

FIND

Description of internal commands

command	option	description
FIND	STRING/K	pattern to search for (STRING)
	COUNT/S	count pattern (doesn't affect cursor position)
	PREV/S	jump to previous occurrence
	NEXT/S	jump to next occurrence
	FIRST/S	jump to first occurrence
	ASK/S	open requester
	CASE/K	case (in)sensitive (BOOL)

QUIET/S don't complain about missing pattern ('not found')

1.187 FIRST

FIRST

Description of internal commands

command	option	description
FIRST	(no options)	move to beginning of line (see GOTO)

1.188 FOLD

FOLD

Description of internal commands

command	option	description
FOLD	OPEN/K	open respectively close fold(s) (BOOL)
	ALL/S	consider all lines
	TOGGLE/S	toggle fold (open/close)

1.189 FORMAT

FORMAT

Description of internal commands

command	option	description
FORMAT	LINES/S	select current paragraphe for formatting
	MARK/S	select block for formatting
	LEFT/S	make selected area left -aligned
	RIGHT/S	make selected area right-aligned
	BLOCK/S	make selected area block-aligned
	CENTER/S	center selected area

1.190 FREEZE

FREEZE

Description of internal commands

command	option	description
---------	--------	-------------

FREEZE	CURRENT/S	freeze current window
	ASK/S	ask for text to unfreeze

1.191 FUNC

FUNC

Description of internal commands

command	option	description
<hr/>		
FUNC	C/S	set mode to C
	BASIC/S	set mode to BASIC
	A68K/S	set mode to Assembler
	PASCAL/S	set mode to Pascal
	AUTODOC/S	set mode to AutoDoc
	STRUCT/S	set mode to C-Header
	CURRENT/S	extract function name from text (below cursor)

Comment: scans text for structures, functions, ... (depending on selected mode) to make up an index. If no mode is specified, the current mode is used.

1.192 FUP

FUP

Description of internal commands

command	option	description
<hr/>		
FUP	(no options)	scroll upwards in fast mode

1.193 GOTO

GOTO

Description of internal commands

command	option	description
<hr/>		
GOTO	LINE/N	line number to go to (ULONG: 1, ...)
	COLUMN/N	column to go to (UWORD: 1, ...)
	UNFOLD/K	unfold if necessary ? (BOOL)
	TOP/S	go to first line of text
	BOTTOM/S	go to last line of text
	OTHEREND/S	toggle position
	CHANGE/S	go to last change

ASK/S	ask for line number to go to
EOL/S	place cursor over last character of line
BFIRST/S	go to first line of marked block
BLAST/S	go to last line of marked block
STEP/N	move cursor left/right (WORD)
TOF/S	move cursor to first line of screen
BOF/S	move cursor to last line of screen

Comment: line numbers are considered to be absolute if UNFOLD is enabled (folded blocks would count as single line otherwise).

1.194 HELP

HELP

Description of internal commands

command	option	description
HELP	CATALOG/K	set database (STRING)
	TOPIC/K	node to look for (STRING)

Comment: if no database (i.e. help file) is specified, the menu's database is used (see config/menus).

1.195 INDENT

INDENT

Description of internal commands

command	option	description
INDENT	ASK/S	open indentation requester
	AUTO/K	set automatic indentation (BOOL)
	SMART/K	set smart indentation (BOOL)
	IN/K	add keyword for smart indentation (STRING)
	OUT/K	add keyword for smart outdentation (STRING)
	CONFIG/K	load smart indentation keyword list (STRING)
	CLR/S	clear smart indentation keyword list

1.196 INFO

INFO

Description of internal commands

command	option	description
---------	--------	-------------

INFO	VERSION/S	show version
	USER/S	show copyright requester
	TEXT/S	show statistics
	ERROR/S	show last error

1.197 INSERT

INSERT

Description of internal commands

command	option	description
INSERT	LINE/S	insert a line
	BLOCK/S	insert into block (see the following options)
	COLUMN/S	insert empty column into block; see AT/N
	STRING/K	insert text into block; see AT/N
	AT/N	column where to insert (UWORD)
	APPEND/S	append text to marked lines

1.198 KEY

KEY

Description of internal commands

command	option	description
KEY	CODE/N	key code to simulate
	RAW/S	make it a rawkey event
	ALT/S	set qualifier ALT
	CTRL/S	set qualifier CONTROL
	SHIFT/S	set qualifier SHIFT

1.199 LAYOUT

LAYOUT

Description of internal commands

command	option	description
LAYOUT	LEFT/N	set left margin for formatting (UWORD)
	RIGHT/N	set right margin for formatting (UWORD)
	WRAP/K	set WordWrap (BOOL)
	ASK/S	open requester to set layout (margin/width)
	CONFIG/K	read layout settings file (STRING)
	AUTO/K	use current indention as left border (BOOL)
	REFORMAT/K	reformat during WordWrpa ? (BOOL)

1.200 LEFT

LEFT

Description of internal commands

command	option	description

LEFT	(no options)	move cursor one position to the left

1.201 LINES

LINES

Description of internal commands

command	option	description

LINES	JOIN/S	join current line & next line
	SWAP/S	swap current line <-> next line
	DOUBLE/S	double current line

1.202 LOCK

LOCK

Description of internal commands

command	option	description

LOCK	CURRENT/S	lock current window
	NAME/K	window to lock (STRING)
	QUIET/S	don't activate window

Comment: To be used within ARexx macros only. An UNLOCK command *must* follow later on to prevent a deadlock situation (take care for a clean exit even after errors using the SIGNAL/OPTION FAILAT commands of ARexx). Exit your script immediately if the LOCK call fails (error code RC is 20). This call doesn't nest: a single UNLOCK unlocks any number of locks.

1.203 MACRO

MACRO

Description of internal commands

command	option	description

MACRO	RECORD/S	toggle sequence recording mode
	PLAY/S	play previously recorded sequence
	LOOPS/N	number of playback loops (UWORD); defaults to 1
	ASK/S	ask for number of loops
	FILE/K	sequence file to load/write (STRING)
	SAVE/S	save previously recorded sequence
	LOAD/S	load a sequence

1.204 MARK

MARK

Description of internal commands

command	option	description

MARK	(no option)	set mark
	HIDE/S	hide mark

1.205 MAXDOWN

MAXDOWN

Description of internal commands

command	option	description

MAXDOWN	(no options)	move to next quarter of document

1.206 MAXUP

MAXUP

Description of internal commands

command	option	description

MAXUP	(no options)	move to previous quarter of document

1.207 MENUS

MENUS

Description of internal commands

command	option	description

MENUS	(no options)	open menu requester

1.208 MISC

MISC

Description of internal commands

command	option	description

MISC	ASK/S	open misc preferences requester
	AUTOBAK/K	set AutoBak mode (BOOL)
	PERIOD/N	set backup interval (UWORD); units are minutes.
	CONFIRM/K	set confirm-backup mode (BOOL)
	PATH/K	set backup path (STRING)
	INFOS/K	set creation of info files (BOOL)
	AUTOLOAD/K	set AutoLoad mode (BOOL)
	AUTOFOLD/K	set AutoFold mode (BOOL)
	CENTERWIN/K	set AutoCenter mode (BOOL)
	FOLDSTART/K	set fold-start marker (STRING)
	FOLDEND/K	set fold-end marker (STRING)
	BACKUP/K	set backup creation (BOOL)

1.209 MODE

MODE

Description of internal commands

command	option	description

MODE	INSERT/K	set insert/overwrite mode (BOOL)
	AUTOCASE/K	set AutoCase mode (BOOL)
	AUTOBRACKET/K	set automatic parenthesis check (BOOL)

1.210 MORE

MORE

Description of internal commands

command	option	description

MORE	(no options)	open further window
	SMART/S	open window if current window is not empty

1.211 NAME

NAME

Description of internal commands

command	option	description

NAME	ASK/S	ask for a new text name
	NEW/F	set new text name (STRING)

1.212 NEW

NEW

Description of internal commands

command	option	description

NEW	(no option)	clear text; user is asked for confirmation
	FORCE/S	clear text at any rate
	NONAME/S	reset name to 'unnamed'

1.213 NEXT

NEXT

Description of internal commands

command	option	description

NEXT	(no options)	move cursor to next word within current line

1.214 OPEN

OPEN

Description of internal commands

command	option	description

OPEN	NAME/M	file(s) to open (STRING or STRINGS)
	FAST/S	use fast loading (no TAB substitution)
	NEW/S	open new window for each file
	AGAIN/S	reload current file
	APPEND/S	append file(s) to current text
	INSERT/S	insert file(s) into current text
	ASK/S	ask for file(s)
	QUIET/S	don't complain about missing files
	PATH/K	default path to be used by file requester (STRING)
	OLDPATH/S	use path of current text as default path
	SMART/S	use current window unless window is not empty

Comment: returns window handle in ARexx mode. The window handle may be used to activate a window later on (see window).

1.215 PATH

PATH

Description of internal commands

command	option	description

PATH	(no options)	open requester to set reference file(s)

1.216 PHRASE

PHRASE

Description of internal commands

command	option	description

PHRASE	CURRENT/S	try to complete current word
	ASK/S	open dictionary requester
	CONFIG/K	load dictionary (STRING)
	ADD/K	add keyword to dictionary (STRING)
	CLR/S	clear dictionary

1.217 PING

PING

Description of internal commands

command	option	description

PING	SLOT/N	write cursor position to named slot

Comment: GoldED offers ten slots ('bookmarks', 0-9) for each window to be recalled by PONG . Slot 0 usually is reserved for use within ARexx scripts.

1.218 PONG

PONG

Description of internal commands

command	option	description

PONG	SLOT/A/N	recall one of 10 bookmark positions (UWORD 0-9)

Comment: GoldED offers ten bookmarks (0-9) for each window. Slot 0 usually is reserved for use within ARexx scripts (see PING).

1.219 POP

POP

Description of internal commands

command	option	description

POP	(no options)	move line from text to pick/push buffer

Comment: the pick/push buffer can hold up to 50 entries (lines). It is a last-in-first-out buffer.

1.220 PREFS

PREFS

Description of internal commands

command	option	description

PREFS	FILE/K	preferences file to load/save (STRING)
	LOAD/S	mode: load from file
	SAVE/S	mode: save to file

1.221 PREV

PREV

Description of internal commands

command	option	description

PREV	(no options)	move cursor to previous word

1.222 PREVEND

PREVEND

Description of internal commands

command	option	description

PREVEND	(no options)	move cursor to end of previous word

1.223 PRINT

PRINT

Description of internal commands

command	option	description

PRINT	FORCE/S	don't ask for confirmation
	BLOCK/S	print block
	ALL/S	print complete file
	LPI/N	set lines per inch value (UWORD)
	LQ/K	set letter quality (BOOL)
	ITALICS/K	set italics printing (BOOL)
	PROP/K	set proportional mode (BOOL)
	DOUBLE/K	set double width mode (BOOL)
	RESET/K	reset printer before output (BOOL)
	CONFIG/K	printer configuration file to load (STRING)
	ASK/S	open printer configuration requester

1.224 PROJECT

PROJECT

Description of internal commands

command	option	description

PROJECT	ASK/S	open project requester

ADD/K	add source file	(STRING)
CONFIG/K	load project definition	(STRING)
DEL/N	remove a source file	(UWORD: 0, ...)
CLR/S	clear list of source files	
LIST/N	set list (struct List *)	

1.225 PUSH

PUSH

Description of internal commands

command	option	description

PUSH	(no options)	insert last line of pick/push buffer into text

1.226 QUERY

QUERY

Description of internal commands

command	option	description

QUERY	NAME/M	variable(s) you are interested in (STRING)
	VAR/K	ARexx variable; where to put the result (STRING).

Comment: Used to query the state of one or more of GoldED's internal variables. This function may be used in interactive mode, too (see misc/command). In interactive mode a requester is used to display the result - e.g. 'LINES=123'. If evoked from a script file (after a LOCK command), the variable name will not be part of the result. In interactive mode it will. In ARexx mode you may specify the name of an ARexx variable to put the result into (defaults to RESULT). The variable name has to be uppercase. It is possible to combine several options (e.g. QUERY DIR DOC); the results will be separated by spaces, too. Valid variable names are:

ABAK..... AutoBackups enabled ? (BOOL)
 ABSLINE..... current absolute line number (ULONG)
 ABSLINES..... absolute number of lines (ULONG)
 ACENTER..... AutoCenter mode (BOOL)
 AFOLD..... AutoFold mode (BOOL)
 ALOAD..... AutoLoad mode (BOOL)
 ANSI..... number of non-ASCII characters (*)
 ANYFOLDS..... does text contain folds ? (BOOL)
 ANYTEXT..... any text in current window ? (BOOL)
 ASKBAK..... ask for backups ? (BOOL)
 BACKUP..... create backups ? (BOOL)
 BAKDIR..... backup path (STRING)

BFIRST..... number of block's 1st line (ULONG). See: MARKED
BITS..... default protection bits (ULONG)
BLAST..... number of block's last line (ULONG). See: MARKED
BOLD..... bold mode used for printing (BOOL)
BRACKET..... automatic parenthesis check ? (BOOL)
BUFFER current line (STRING)
BYTES..... text size (number of bytes) (ULONG) (*)
CAT..... catalog name/language (STRING)
CHKCASE..... automatic case check ? (BOOL)
CODE..... ASCII code of character under cursor (UBYTE)
COLUMN..... current column (UWORD: 1, ...)
COLUMNS..... window width (UWORD)
CON..... window dimension string (STRING)
CURRENT..... pointer to memory area of current line (char *)
DIR..... path used by file requester (STRING)
DOC..... window title = file name (STRING)
DOUBLE..... use doublestrike printing ? (BOOL)
DTABS..... use distinctive TABs ? (BOOL)
ERR..... last error (STRING)
FILE..... name of current text without path (STRING)
FIND..... search pattern (STRING)
FOLDA..... fold start marker (STRING)
FOLDB..... fold end marker (STRING)
FOLDS..... number of folds in text (ULONG) (*)
FONTX..... width of text font (UWORD)
FONTY..... height of text font (UWORD)
FUNC..... mode of QuickFunc list (UWORD: 1, ...)
HANDLE..... window handle of current window (ULONG)
HMI..... horizontal motion index/printer (UWORD)
HOST..... name of ARexx port (STRING)
INDENT..... AutoIndention ? (BOOL)
INFOS..... create *.info files ? (BOOL)
INSERT..... insert mode used ? (BOOL)
ITALICS..... italics printing ? (BOOL)
LEFT..... layout: left margin (UWORD)
LEN..... length of current line
LINE..... number of current line; not absolute (ULONG)
LINES..... number of lines; not absolute (ULONG)
LPI..... lines per inch/printer (UWORD)
LQ..... letter quality printing ? (BOOL)
MARKED..... any block marked ? (BOOL)
MAXLEN..... length of longest line (UWORD) (*)
MODIFY..... has text been modified ? (BOOL)
PATH..... path of current text (STRING)
PICKED..... number of lines in pick/push buffer (UWORD)
PROG..... program's name - usually GoldED (STRING)
PRJLIST list of project files; see misc/project setup
READONLY..... is window read-only ? (BOOL)
REM..... (file-)comment of current text (STRING)
REMAP..... character translation file (STRING)
RESET..... reset printer before output ? (BOOL)
RIGHT..... layout: right margin (UWORD)
ROWS..... window height (UWORD)
RPLC..... replace text (STRING)
SCREEN..... screen name (STRING)
SCRMODE..... screen mode ID (ULONG)
SCRTYPE..... screen type: public/custom (UWORD)

```

SHANGHAI..... shanghai mode set ? (BOOL)
SINDENT..... smart indentation enabled ? (BOOL)
SOLID..... solid TABs ? (BOOL)
STDLINE..... is current line a 'normal' line ? (BOOL)
TAB..... TAB distance keyboard (UWORD)
TABFILE..... TAB distance for TAB substitution (UWORD)
TIMER..... backup interval - minutes (UWORD)
TOPLINE..... line number of window's first line (ULONG)
UNFOLD..... unfold during GOTO ? (BOOL)
USECASE..... case sensitive search/replace ? (BOOL)
VER..... version string (STRING)
WINDOWS..... number of open text windows (UWORD)
WINH..... window height (UWORD)
WINW..... window width (UWORD)
WORDS..... number of words (ULONG) (*)
WORD..... current word (STRING)
WRAP..... word wrap mode used ? (BOOL)
X..... window's left edge (UWORD)
XPK..... XPK compression mode (STRING)
Y..... window's top edge (UWORD)

```

(*): These variables are valid after an EXAll command only.

1.227 QUIT

QUIT

Description of internal commands

command	option	description
QUIT	(no option)	close current window (ask for confirmation)
	FORCE/S	close current window at any rate
	UNLOAD/S	close window, unload editor (see HotKey)

1.228 REFRESH

REFRESH

Description of internal commands

command	option	description
REFRESH	PAGE/S	redraw current text
	LINE/S	redraw current line

1.229 REMAP

REMAP

Description of internal commands

command	option	description

REMAP	TABLE/K	remap text; use this translation file (STRING)
	ASK/S	open remap requester

1.230 REPLACE

REPLACE

Description of internal commands

command	option	description

REPLACE	STRING/K	pattern to look for (STRING)
	BY/K	replacement for pattern (STRING)
	PREV/S	replace previous occurrence of pattern (*)
	NEXT/S	replace next occurrence of pattern only (*)
	ALL/S	replace all occurrences of pattern (*)
	BLOCK/S	replace pattern within marked lines (*)
	ASK/S	open find/replace requester
	CASE/K	set case sensitive mode (BOOL)
	QUIET/S	don't complain about missing pattern ('not found')

(*) only one of these options may be specified.

1.231 REQUEST

REQUEST

Description of internal commands

command	option	description

REQUEST	HIDE/K	turn requesters off
	DEFAULT/K	set default if requesters are off (UWORD)
	BODY/K	body text, lines separated by ' ' (STRING)
	BUTTON/K	text for button(s), separated by ' ' (STRING)
	TITLE/K	requester title (STRING)
	LONG/S	ask for a number
	MIN/N	set lower limit for input of number (WORD)
	MAX/N	set upper limit for input of number (WORD)
	OLD/K	default value (STRING)
	FILE/S	ask for a file
	PATH/K	default path if asking for a file (STRING)
	VAR/K	ARexx variable; where to put the result (STRING)
	STRING/S	ask for a string

STATUS/K text to display in status line (STRING)

Comment: Don't use the hide option outside of ARexx macros. Turn requesters on again at the end of your macro. Turning requesters off is useful to suppress requesters like 'pattern not found' (wich would annoy the user); the return code RC is checked instead to determine wether a command has completed successfully.

1.232 RIGHT

RIGHT

Description of internal commands

command	option	description
RIGHT	(no options)	move cursor one position to the right

1.233 RX

RX

Description of internal commands

command	option	description
RX	CMD/K	command to be send to ARexx server/other program
	ASK/S	ask for command
	PORT/K	where to send the command (defaults to "AREXX")
	MACRO/K	macro to execute if PORT is not valid

Comment: The macro is called with the command string as first argument. Basically same usage as the AmigaDos command RX if you set the port to AREXX: Use single quotations marks to have the command string evaluated by the Arexx server (e.g. rx 'info user'). Use double quotation marks or no quotation marks at all to make the server execute a script (e.g. rx golded:arexx/empty.ged).

1.234 SAVE

SAVE

Description of internal commands

command	option	description

SAVE	BLOCK/S	save block only
	ALL/S	save complete file
	NAME/K	set file name to be used for saving
	ASK/S	open save-as requester
	EXIT/S	exit GoldED if save operation is succesful
	CRUNCH/S	compress file (XPK) while saving it
	XPKMODE/K	set XPK compression mode (STRING, e.g. IMPL)

1.235 SCREEN

SCREEN

Description of internal commands

command	option	description

SCREEN	ASK/S	open display mode requester
	USE/K	name of public screen to use (STRING)
	CONFIG/K	load this display configuration file (STRING)
	ICONIFY/K	(un)iconify (BOOL)
	FRONT/S	move GoldED's screen to the front
	BACK/S	move GoldED's screen to the back
	CLOSE/S	close screen (wait for HotKey activation)

1.236 SHIFT

SHIFT

Description of internal commands

command	option	description

SHIFT	COLUMNS/N	number of columns to shift (UWORD)
	TAB/S	set shifting distance to TAB size
	ASK/S	open requester (left/right shifting)
	LEFT/S	shift to the left
	RIGHT/S	shift to the right (indent)

1.237 SMARTCR

SMARTCR

Description of internal commands

command	option	description

SMARTCR	(no options)	'smart' return (no splitting of line)

1.238 TAB

TAB

command	option	description
TAB	BACK/S	perform a backtab (else a normal tab)
	SOLID/K	make it a solid TAB (BOOL)
	FIXED/S	make it a distinctive TAB
	DYNAMIC/S	make it a dynamic TAB (see dynamic TABs)
	REGULAR/S	make it a regular (standard) TAB

Comment: options of this command may be used to change TAB mode temporarily. They do not affect global TAB settings (use TABS instead).

1.239 TABS

TABS

Description of internal commands

command	option	description
TABS	FIXED/S	set TAB mode to distinctive TAB's
	REGULAR/S	set TAB mode to regular TAB's
	DYNAMIC/S	set TAB mode to dynamic TAB's
	FILE/N	number of SPC's used for TAB substitution (UWORD)
	KEY/N	TAB distance on screen
	ASK/S	open TAB preferences requester
	SET/N	add distinctive TAB position (UWORD)
	CLR/S	clear all distinctive TAB positions
	SOLID/K	make TABs solid (BOOL)

1.240 TASK

TASK

Description of internal commands

command	option	description
TASK	PRI/N	set task priority of GoldED (WORD, -3 to 3)
	DEBUG/K	set debug mode (BOOL)

Comment: in debug mode recieved ARexx commands are printed to standard output.

1.241 TEXT

TEXT

Description of internal commands

command	option	description

TEXT	T/K	text to be inserted at cursor position (STRING)
	STAY/S	don't move cursor while inserting text
	CR/S	append linefeed to text

Comment: Use may use "*" with the text string to insert quotation marks (single quotation marks are considered as string delimiters). Example usage:

TEXT T="He said *Bye !*" and left the room."

1.242 UJUMP

UJUMP

Description of internal commands

command	option	description

UJUMP	(no options)	jump to beginning of screen / previous page

Comment: Cursor jumps to first line of screen if placed below that line so far. Jumps to previous page if placed in line one already. Compare: UPAGE .

1.243 UNDO

UNDO

Description of internal commands

command	option	description

UNDO	(no options)	undo line (restore previous contents)

1.244 UNLOCK

UNLOCK

Description of internal commands

command	option	description
---------	--------	-------------

UNLOCK	(no option)	unlock GUI after a previously call to LOCK
DELAY/S		unlock GUI, delay until exit of GoldED
STICKY/S		unlock GUI, delay until current window is closed

Comment: The DELAY/STICKY options are reserved for use by external applications (e.g. the QuickStarter ED). They provide ways and means to synchronize with GoldED.

1.245 UP

UP

Description of internal commands

command	option	description
<hr/>		
UP	(no options)	move cursor one line up

1.246 UPAGE

UPAGE

Description of internal commands

command	option	description
<hr/>		
UPAGE	(no options)	move to previous page (compare UJUMP)

1.247 USE

USE

Description of internal commands

command	option	description
<hr/>		
USE	(no options)	accept current line

Comment: To be used within ARexx macros only. After having written directly to the memory area of the current line (which is *dangerous*) you have to call this function to make GoldED accept your changes. Use QUERY CURRENT (see QUERY) to get a pointer to the current line's buffer. It is not possible to change the lenght of the current line by poking into the line buffer.

1.248 VIEW

VIEW

Description of internal commands

command	option	description

VIEW	LEFT/S	shift view left
	RIGHT/S	shift view right
	COLUMNS/N	number of columns to shift (UWORD)

1.249 WINDOW

WINDOW

Description of internal commands

command	option	description

WINDOW	MAX/S	blow up current window
	CENTER/S	center current window on screen
	ARRANGE/S	arrange windows on screen
	ZIP/S	zip window
	USE/K	activate named window/file (STRING)
	FORCE/S	load named file if necessary (see USE/K)
	WIDTH/N	resize window width (UWORD)
	HEIGHT/N	resize window height (UWORD)
	X/N	set window's x position (UWORD)
	Y/N	set window's y position (UWORD)
	NEXT/S	activate next window
	RECOVER/S	redraw window
	HANDLE/N	activate window using its handle (ULONG)
	ORDINAL/N	activate 1st , 2nd, ... window (ULONG: 0, ...)

Comment: A window handle is returned by the OPEN function. You may query the current window's handle, too (see QUERY /handle).

1.250 XREF

XREF

Description of internal commands

command	option	description

XREF	CURRENT/S	open help text related to current word
	PHRASE/K	open help text related to this phrase (STRING)
	ASK/S	ask for topic

1.251 GENERAL HINTS

GENERAL HINTS

Never change display mode if you are short of memory - the editor might not be able to reopen screen/windows.

Turn AutoFold (see config/misc) OFF if you do not need folding. This will speed up loading since the editor won't have to examine each line after loading a text from disk (looking for fold markers).

Use the same marker strings for beginning/end of folded sections (see config/misc) to increase the speed of folding: the editor won't have to look for two different strings. However, using the same strings doesn't allow nested folding.

The backup path should point to the same device as your main text directory, otherwise backup generation is slowed down because file copies have to be used instead of a fast DOS rename (which is not available accross devices).

If you have accidentally cleared a text you could try to restore it using the provided recover utility (GoldED:tools/recover drawer).

Don't use soft/hardlinks with text files. If a file is renamed by the editor (happens during backup creation) the link doesn't change, i.e. it would point to the backup instead of the new file.

Do not use fast loading if you are short of RAM or for huge files since fast loading requires a huge buffer of exactly the original file's size. Attempting to load a 400 KB file would allocate a 400 KB IO buffer as well as about 450 KB to store the text (however the 400 KB IO buffer is freed immediately after loading). Slow loading is much more efficient in terms of memory consumption: a 16K buffer is required, no matter how large a file is.

Most file requesters of GoldED do support multi select. You could use this feature to join a couple of files together using either project/append or project/insert .

GoldED as well as any Amiga shell does employ the clipboard for cut and paste operations. You can use this feature to insert a directory dump from a shell window into your current document. Or (vice versa) cut a line from your text & paste it to a shell window (doesn't make that much sense though :-).

The editor itself is not able to read the position of the 'sleep' icon (reading an AppIcon's position is not supported by the OS), so it is not able to remember the icon's position after you have moved it. Nevertheless it is possible to snapshot the preferred position: Open the 'prefs/env-archive/GoldED' directory. Move the 'AppIcon' icon from within that directory to desired location, than snapshot it (icon menu of workbench). Finally move the icon back to its drawer. The editor will read the new position the next time it is evoked.

The Amiga OS does support font/library caching: Fonts or libraries opened by GoldED usually are not removed from memory when GoldED is closed - instead they are marked as 'unused'. The OS is able to flush these ressources if required (in low memory situations), so the memory occupied by libs/fonts is

not lost (you may verify this by using the 'avail flush' command).

If you own a fast Amiga (4000), operations like scrolling (especially in monochrome mode) are hardly limited in speed by GoldED itself: They almost solely depend on key repeat rate settings. In other words: use the input preferences program of your workbench to speed up GoldED by increasing the key repeat rate.

In spite of the fact that GoldED does keep its default configuration file in 'envarc:GoldED' - do not use this directory to store further configuration files: A copy of envarc is written to RAM: during startup; you would waste memory. 'GoldED:' or even 's:' are more appropriate locations.

1.252 CREDITS

CREDITS

* DICE * Reqtools * XPK * ARexxBox * GadToolsBox *

This program has been developed using Matt Dillon's Dice C compiler (v2.07.54R). Most of the requester design has been done using Jan van den Baard's 'GadToolsBox'. Thanx to Nico François for his 'reqtools.library'. Further credits got to the developers of 'xpk.library' - who created a fine (de)compression standard. The ARexx routines of GoldED are somewhat related to source code created by Michael Balzer's ARexxBox (though less sophisticated). Thanks to Joerg Gutzke & Dario Fava for running Mowgli BBS respectively Mailway BBS (where I got most of the stuff above).

GUIMake has been developed by Rico Krasowski. Included with kind permission of the author. Thanks a lot for this fine program, Rico !

1.253 HOW TO REGISTER

HOW TO REGISTER

If you are currently using the unregistered version (saving/printing limited to 1000 lines) you may want to register. The registered GoldED consists of a mailing of one or more disks. Please allow up to 8 weeks for delivery. This is the worst case. Average is one week. To register for GoldED, send an euro(!)check to me (see How to contact author). Cash is fine, too (Europe: must be DM, outside Europe: must be US-\$). Please write clearly or type. The following packages are available (offers valid until end of February 1994):

GoldED Light (20 DM Europe, 20\$ outside Europe)

Mailing consists of one or two disks. Includes single user licence for GoldED (except HiSpeed). The latest versions of registered GoldED as well as standard documentation (disk based) is shipped.

GoldED Pro (30 DM Germany, 35 DM Europe, 30\$ outside Europe)

Mailing consists of two disks and a printed manual. Includes single user licence for the latest versions of GoldED & HiSpeed.

In generally FD support tools and libraries are not part of the registered distribution (even if they come along with the unregistered one) unless I get permission from the authors. I do provide disk(s)/postage. Special conditions are available if you order more than one copy. Don't forget to provide a user name/address for each of them (required for registration):

2 packages -20% each
more (just joking :-) -30% each

1.254 HOW TO GET UPDATES

HOW TO GET UPDATES

The only (I repeat: only) way to receive updates, whether registered or unregistered, is to call your local BBS and look out for the latest GoldED copy. Registered users may use the unlock utility to turn unregistered versions into registered ones. Don't send me disks unless you want to get rid of them. My uploads usually go to Mowgli BBS (Aachen, Germany) and Mailway BBS (Aachen, Germany):

MOWGLI BBS

FidoNet 2:242/7, phone: 49-(0)241-40 59 49 (V22, V22bis, V32, V32bis, V42, V42bis, HST, MNP). Online: 0.00 - 24.00. You may use the magic GoldED for requesting the GoldED package (about 550K, e.g. 'GED093.lha').

MAILWAY BBS

Z-Net, phone: 49-(0)2405-74863 (V22, V22bis, V32, V32bis, V42, V42bis, ZyXel 16.8K, MNP). Online: 0.00 - 24.00.

1.255 HOW TO CONTACT AUTHOR

HOW TO CONTACT AUTHOR

Feel free to e-mail me bug reports, comments or suggestions. Current e-mail address is DIETMAR@MAILWAY.ZER (German Z-Net). FidoNet users may write to Mowgli 2:242/7, too (but since I don't run a Fido point, I won't be able to answer). Do not send normal letters unless you want to register - at least do not expect me to answer. You definitely won't get an answer if you fail to provide a self-addressed envelope, postage paid. In generally you can reach me by writing to one of the addresses below. These addresses are ****not**** valid for requesting GoldED updates (see: How to get updates).

Dietmar Eilert
 Mies-v-d-Rohe-Str. 31
 52074 Aachen (Germany)
 Phone: 0241/81665

My second address (not valid when I wrote these lines):

Dietmar Eilert
 Kampstrasse 28
 59269 Beckum (Germany)
 Phone: 02525/7776

Better give me a ring to find out which one is valid when you read this text.
 If you decide for the wrong one your letter is delayed for about one month.

1.256 GoldED

???	FIND MENU	MENUS
APC	FIND/ASCII INSERT	MISC
APPICON	FIND/ASCII TABLE	MISC MENU
APPWINDOWS	FIND/CHARACTER SET	MISC/CALCULATOR
AREXX PORT	FIND/CHECK	MISC/COMMAND
AREXXBOX	FIND/COMPLETE	MISC/DELETE FILE
ARGUMENTS	FIND/COUNT	MISC/DOUBLE LINE
AUTOBACKUP	FIND/FIND	MISC/HELP
AUTOCASE	FIND/FIND NEXT	MISC/HISPEED
AUTOCENTER	FIND/FIND PREVIOUS	MISC/INSERT DATE
AUTOFOLD	FIND/FUNCTIONS	MISC/INSERT TIME
AUTOINDENTION	FIND/INSERT CODE ESC	MISC/LAST ERROR
AUTOLOAD	FIND/INSERT CODE FF	MISC/NEW SHELL
BACK	FIND/MATCHING BRACKET	MISC/PICK LINE
BIND	FIND/REFERENCE	MISC/PROJECT SETUP
BITS	FIND/REFERENCE...	MISC/PUSH LINE
BLOCK	FIND/REPLACE	MISC/STATISTICS
BLOCK MENU	FIND/REPLACE NEXT	MISC/SWAP LINES
BLOCK/APPEND TEXT	FIND/SHOW CODE	MISC/UNDO LINE
BLOCK/BCOPY	FIRST	MODE
BLOCK/BDELETE	FOLD	MORE
BLOCK/BMOVE	FOLDING	MULTIPLE COMMANDS
BLOCK/COLUMN TEXT	FONT	MULTISELECT
BLOCK/COPY	FORMAT	NAME
BLOCK/CUT	FORMATTER	NEW
BLOCK/DELETE COLUMN	FREEZE	NEXT
BLOCK/HIDE MARK	FUNC	OPEN
BLOCK/INDENT	FUP	PARAGRAPHE VS. BLOCK
BLOCK/INSERT COLUMN	GADTOOLSBOX	PARENTHESIS CHECK
BLOCK/LOWERCASE	GENERAL HINTS	PATH
BLOCK/MARK	GETTING STARTED	PHRASE
BLOCK/PASTE	GOTO	PING
BLOCK/PRINT	GUI	PONG
BLOCK/SAVE AS	GUIMAKE	POP
BLOCK/SORT	HELP	PREFS
BLOCK/UPPERCASE	HELP KEY	PREV
BORDERLESS	HOTKEY	PREVEND

BRACKET	HOW TO CONTACT AUTHOR	PRINT
CHARACTER SET REMAP	HOW TO FOLD LINES	PROJECT
CLIP	HOW TO GET UPDATES	PROJECT MENU
CLIPBOARD	HOW TO REGISTER	PROJECT/ABOUT
CMD	INDENT	PROJECT/APPEND
CODE	INFO	PROJECT/BITS
COLON	INSERT	PROJECT/CLEAR TEXT
COMMAND LIST	INSERTION OF COLUMNS	PROJECT/CURRENT DIR
CONFIG	INTERNAL COMMANDS	PROJECT/INSERT
CONFIG MENU	INTRODUCTION	PROJECT/LOAD
CONFIG/CALENDER-TIME	KEY	PROJECT/LOAD FAST
CONFIG/Dictionary	KEYBOARD	PROJECT/LOAD NEW
CONFIG/DISPLAY	LANGUAGE	PROJECT/LOAD ORIGINAL
CONFIG/INDENTION	LAYOUT	PROJECT/MORE ED
CONFIG/KEYBOARD BINDINGS	LAYOUT MENU	PROJECT/NEW NAME
CONFIG/LOAD	LAYOUT/BLOCK CENTER	PROJECT/PRINT
CONFIG/MENUS	LAYOUT/BLOCK LEFT	PROJECT/QUIT
CONFIG/MISC	LAYOUT/BLOCK LEFT/RIGHT	PROJECT/SAVE
CONFIG/PRINTER	LAYOUT/BLOCK RIGHT	PROJECT/SAVE & EXIT
CONFIG/REFERENCES	LAYOUT/BORDERS	PROJECT/SAVE AS
CONFIG/SAVE	LAYOUT/SECTION BLOCK	PROJECT/SAVE AS XPK
CONFIG/TABS	LAYOUT/SECTION CENTER	PROJECT/USER
CONFIGURATION GUIDELINES	LAYOUT/SECTION LEFT	PUSH
CONTROL MENU	LAYOUT/SECTION RIGHT	QUERY
CONTROL/ARRANGE WINDOWS	LAYOUT/WORD WRAP	QUICKFUNC
CONTROL/CENTER WINDOW	LEFT	QUICKREFERENCE
CONTROL/ENLARGE WINDOW	LICENCE	QUICKSTARTER
CONTROL/FOLD ALL	LINES	QUIT
CONTROL/FREEZE WINDOW	LOCK	REFRESH
CONTROL/GO TO LINE	LOCK A WINDOW	REMAP
CONTROL/ICONIFY	MACRO	REMOVAL OF COLUMNS
CONTROL/INSERT-OVERWRITE	MACRO MENU	REPLACE
CONTROL/NEXT WINDOW	MACRO RECORDING	REQTOOLS
CONTROL/RECALL POSITION	MACROS/COMPILE & LINK	REQUEST
CONTROL/STORE POSITION	MACROS/COMPILE & LINK NEW	REQUIRED SYSTEM
CONTROL/TO LAST CHANGE	MACROS/COMPILE+LINK TEXT	RETURN KEY
CONTROL/TOGGLE TAB MODE	MACROS/CONFIG	RIGHT
CONTROL/TOP-BOTTOM	MACROS/DISPLAY ERRORS	RX
CONTROL/UNFOLD ALL	MACROS/DMAKE	SAVE
CONTROL/UNFREEZE WINDOW	MACROS/EDIT MACRO	SCREEN
CONTROL/ZIP WINDOW	MACROS/ERROR: FIRST	SELECT A HOST
CR	MACROS/ERROR: NEXT	SHIFT
CREDITS	MACROS/ERROR: PREVIOUS	SHIFTING
CURSOR KEYS	MACROS/GUIMAKE	SMARTCR
DEL	MACROS/LOAD	SMARTINDENTION
DEL KEY	MACROS/MODULES	SPEEDS OF SCROLLING
DELETE	MACROS/NUMBER	STARTUP MACRO
DESCRIPTION OF MENUS	MACROS/OPTIONS	TAB
DICE	MACROS/PLAY	TAB KEY
DIR	MACROS/RECORD START-STOP	TABS
DISPLAY MODE	MACROS/REVISION	TASK
DJUMP	MACROS/RUN EXECUTABLE	TEXT
DO YOUR JOB	MACROS/RUN TEXT AS MACRO	UJUMP
DOWN	MACROS/SAVE	UNDO
DPAGE	MAGIC CODES	UNLOCK
DYNAMIC TABS	MAIN	UNLOCK GUI
ENDWORD	MARK	UP

ESC KEY	MAXDOWN	UPAGE
EVENT	MAXUP	USE
EVENT DEFINITION	MENU TREE OF BLOCK MENU	VIEW
EXALL	MENU TREE OF CONFIG MENU	WINDOW
F-KEYS	MENU TREE OF CONTROL MENU	WORDWRAP
FASTLOAD	MENU TREE OF FIND MENU	XPK
FDOWN	MENU TREE OF LAYOU MENU	XPK SUPPORT
FEATURE LIST	MENU TREE OF MACRO MENU	XREF
FILE	MENU TREE OF MISC MENU	
FIND	MENU TREE OF PROJECT MENU	
FIND MENU	MENUHELP	
