

**GED.HYPER**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> GED.HYPER	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		March 29, 2025
<i>SIGNATURE</i>		

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>GED.HYPER</b>	<b>1</b>
1.1	main . . . . .	1
1.2	FEATURES . . . . .	1
1.3	LICENCE . . . . .	2
1.4	INTRODUCTION . . . . .	3
1.5	REQUIRED SYSTEM . . . . .	3
1.6	GETTING STARTED . . . . .	4
1.7	MOUSE CONFIGURATION . . . . .	6
1.8	MENU . . . . .	6
1.9	PROJECT MENU . . . . .	7
1.10	project/clear window . . . . .	7
1.11	project/open . . . . .	7
1.12	project/open fast . . . . .	8
1.13	project/open new . . . . .	8
1.14	project/open original . . . . .	9
1.15	project/insert . . . . .	9
1.16	project/append . . . . .	9
1.17	project/save . . . . .	9
1.18	project/save as . . . . .	10
1.19	project/save as XPK . . . . .	10
1.20	project/save and close . . . . .	10
1.21	project/print . . . . .	10
1.22	project/name . . . . .	11
1.23	project/information . . . . .	11
1.24	project/protection . . . . .	11
1.25	project/current dir... . . . .	12
1.26	project/current dir . . . . .	12
1.27	project/quit all & exit . . . . .	12
1.28	project/quit buffer . . . . .	12
1.29	BLOCK MENU . . . . .	13

---

---

1.30	block/mark	13
1.31	block/mark line	14
1.32	block/mark paragraph	14
1.33	block/mark hide	14
1.34	block/clipboard cut	14
1.35	block/clipboard copy	14
1.36	block/clipboard paste	15
1.37	block/paste column	15
1.38	block/copy	16
1.39	block/delete	16
1.40	block/move	16
1.41	block/column	16
1.42	block/column delete	16
1.43	block/column insert	17
1.44	block/column text	17
1.45	block/append text	17
1.46	block/block sort	17
1.47	block/block to uppercase	18
1.48	block/block to lowercase	18
1.49	block/block save	18
1.50	block/block print	18
1.51	LAYOUT MENU	18
1.52	layout/set right margin	19
1.53	layout/use current margin	19
1.54	layout/insert mode	19
1.55	layout/solid tabs	20
1.56	layout/word wrap	20
1.57	layout/templates	20
1.58	layout/AutoCase	20
1.59	layout/block justified	21
1.60	layout/block left	21
1.61	layout/block right	21
1.62	layout/block centered	21
1.63	layout/indent	22
1.64	layout/indent block	22
1.65	layout/indent line	22
1.66	layout/paragraph justified	22
1.67	layout/paragraph left	22
1.68	layout/paragraph right	23

---

---

1.69 layout/paragraph centered . . . . .	23
1.70 layout/NumPad = movement . . . . .	23
1.71 SEARCH MENU . . . . .	23
1.72 search/find . . . . .	24
1.73 search/find next . . . . .	24
1.74 search/find previous . . . . .	24
1.75 search/replace . . . . .	25
1.76 search/replace next . . . . .	25
1.77 search/count pattern . . . . .	25
1.78 search/check . . . . .	25
1.79 search/matching bracket . . . . .	26
1.80 search/reference . . . . .	26
1.81 search/reference... . . . .	26
1.82 search/complete . . . . .	27
1.83 search/ASCII table . . . . .	27
1.84 search/ASCII code ? . . . . .	27
1.85 search/ASCII code insert . . . . .	28
1.86 search/character set . . . . .	28
1.87 search/function list . . . . .	28
1.88 search/function find . . . . .	29
1.89 VIEW MENU . . . . .	29
1.90 view/new window . . . . .	30
1.91 view/go to line . . . . .	30
1.92 view/go to top-bottom . . . . .	30
1.93 view/go to modification . . . . .	30
1.94 view/position store . . . . .	30
1.95 view/position recall . . . . .	31
1.96 view/fold . . . . .	31
1.97 view/preview . . . . .	32
1.98 view/syntax highlighting . . . . .	32
1.99 view/debugger . . . . .	32
1.100view/toolbar . . . . .	33
1.101view/window . . . . .	33
1.102view/window arrange . . . . .	34
1.103view/window maximize . . . . .	34
1.104view/window center . . . . .	34
1.105view/window zip . . . . .	34
1.106view/remember size . . . . .	35
1.107view/window next . . . . .	35

---

---

1.108view/window previous . . . . .	35
1.109view/hidden buffers . . . . .	35
1.110EXTRAS MENU . . . . .	36
1.111extras/project filter . . . . .	36
1.112extras/search file . . . . .	37
1.113extras/search file... . . . . .	37
1.114extras/line swap . . . . .	37
1.115extras/line double . . . . .	37
1.116extras/line remove . . . . .	38
1.117extras/line insert . . . . .	38
1.118extras/undo . . . . .	38
1.119extras/redo . . . . .	38
1.120extras/AmigaDOS shell . . . . .	39
1.121extras/HiSpeed . . . . .	39
1.122extras/files . . . . .	39
1.123extras/insert text . . . . .	39
1.124extras/options local . . . . .	40
1.125extras/options local/project . . . . .	41
1.126extras/options local/hunter . . . . .	41
1.127extras/options local/reference . . . . .	41
1.128extras/options local/printer . . . . .	42
1.129extras/options local/words . . . . .	42
1.130extras/options local/templates . . . . .	43
1.131extras/options local/menu . . . . .	43
1.132Event definition . . . . .	44
1.133extras/options local/mouse . . . . .	45
1.134extras/options local/keyboard . . . . .	45
1.135extras/options local/misc . . . . .	46
1.136extras/options local/API . . . . .	47
1.137extras/options local/tabs . . . . .	48
1.138extras/options local/layout . . . . .	49
1.139extras/options local/GUI . . . . .	49
1.140extras/options local/display . . . . .	50
1.141extras/options local/syntax . . . . .	52
1.142extras/options local/gadgets . . . . .	53
1.143extras/options local/indent . . . . .	53
1.144extras/options global . . . . .	53
1.145extras/options global/filetypes . . . . .	54
1.146extras/options global/misc . . . . .	54

---

---

1.147	extras/options global/variables	57
1.148	extras/options global/gui	57
1.149	extras/save preferences	58
1.150	MACRO MENU	58
1.151	macros/session load	58
1.152	macros/session save	59
1.153	macros/macro edit	59
1.154	macros/macro start	59
1.155	macros/macro execute	60
1.156	macros/sequence record	60
1.157	macros/sequence load	60
1.158	macros/sequence save	61
1.159	macros/sequence play	61
1.160	macros/sequence play...	61
1.161	macros/macros	61
1.162	macros/execute command	62
1.163	macros/execute line	62
1.164	About GoldED	63
1.165	Keyboard	63
1.166	Cursor keys	63
1.167	HELP key	64
1.168	TAB key	65
1.169	RETURN key	65
1.170	DEL key	65
1.171	ESC key	66
1.172	SPACE key	66
1.173	F-Keys	67
1.174	ARexx port	67
1.175	Select a host	68
1.176	Lock a window	68
1.177	Do your Job	69
1.178	Unlock GUI	69
1.179	Internal commands	70
1.180	Command list	70
1.181	API	71
1.182	BACK	71
1.183	BEEP	71
1.184	BIND	72
1.185	BITS	72

---

---

1.186BLOCK . . . . .	72
1.187BRACKET . . . . .	73
1.188BREAKPT . . . . .	73
1.189CLIP . . . . .	73
1.190CMD . . . . .	74
1.191CODE . . . . .	74
1.192COLON . . . . .	74
1.193CR . . . . .	75
1.194DEBUG . . . . .	75
1.195DEL . . . . .	75
1.196DELETE . . . . .	75
1.197DIR . . . . .	76
1.198DJUMP . . . . .	76
1.199DOWN . . . . .	76
1.200DPAGE . . . . .	77
1.201ELSE . . . . .	77
1.202ENDIF . . . . .	77
1.203ENDWORD . . . . .	77
1.204EXALL . . . . .	77
1.205EXTRACT . . . . .	78
1.206FDOWN . . . . .	78
1.207FILE . . . . .	78
1.208FIND . . . . .	79
1.209FIRST . . . . .	79
1.210FIX . . . . .	79
1.211FOLD . . . . .	80
1.212FORMAT . . . . .	80
1.213FREEZE . . . . .	80
1.214FUNC . . . . .	80
1.215FUP . . . . .	81
1.216GOTO . . . . .	81
1.217GREP . . . . .	82
1.218GUI . . . . .	82
1.219HELP . . . . .	82
1.220HUNTER . . . . .	82
1.221IF . . . . .	83
1.222INDENT . . . . .	83
1.223INFO . . . . .	83
1.224INSERT . . . . .	84

---

---

1.225KEY . . . . .	84
1.226LAYOUT . . . . .	84
1.227LEFT . . . . .	84
1.228LINES . . . . .	85
1.229LOCK . . . . .	85
1.230MACRO . . . . .	85
1.231MARK . . . . .	86
1.232MAXDOWN . . . . .	86
1.233MAXUP . . . . .	86
1.234MENUS . . . . .	87
1.235MISC . . . . .	87
1.236MORE . . . . .	87
1.237MOUSE . . . . .	87
1.238NAME . . . . .	88
1.239NEW . . . . .	88
1.240NEXT . . . . .	88
1.241NOTIFY . . . . .	88
1.242OPEN . . . . .	89
1.243PATH . . . . .	89
1.244PC . . . . .	89
1.245PHRASE . . . . .	90
1.246PING . . . . .	90
1.247PONG . . . . .	90
1.248POP . . . . .	91
1.249PREFS . . . . .	91
1.250PREV . . . . .	91
1.251PREVEND . . . . .	92
1.252PRINT . . . . .	92
1.253PROJECT . . . . .	92
1.254PUSH . . . . .	92
1.255QUERY . . . . .	93
1.256QUIT . . . . .	95
1.257REDO . . . . .	95
1.258REFRESH . . . . .	96
1.259REMAP . . . . .	96
1.260REPLACE . . . . .	96
1.261REQLIST . . . . .	97
1.262REQUEST . . . . .	97
1.263RIGHT . . . . .	97

---

---

1.264RUN . . . . .	98
1.265RX . . . . .	98
1.266SAVE . . . . .	98
1.267SCREEN . . . . .	99
1.268SESSION . . . . .	99
1.269SET . . . . .	100
1.270SHIFT . . . . .	100
1.271SMARTCR . . . . .	100
1.272STOP . . . . .	100
1.273SUFFIX . . . . .	101
1.274SYNTAX . . . . .	101
1.275TAB . . . . .	101
1.276TABS . . . . .	101
1.277TASK . . . . .	101
1.278TEXT . . . . .	102
1.279TMPLATE . . . . .	102
1.280TYPE . . . . .	102
1.281UJUMP . . . . .	103
1.282UNDO . . . . .	103
1.283UNLOCK . . . . .	103
1.284UP . . . . .	103
1.285UPAGE . . . . .	104
1.286USE . . . . .	104
1.287VIEW . . . . .	104
1.288VLEFT . . . . .	105
1.289VRIGHT . . . . .	105
1.290WINDOW . . . . .	105
1.291WORD . . . . .	106
1.292XREF . . . . .	106
1.293Input events . . . . .	106
1.294TOOLS . . . . .	107
1.295HiSpeed . . . . .	107
1.296HARDWARE REQUIREMENTS . . . . .	108
1.297CONFIGURATION . . . . .	108
1.298PAGE HEADER . . . . .	110
1.299PRINTING . . . . .	110
1.300SETTINGS & UNITS . . . . .	111
1.301PAGE LAYOUT . . . . .	111
1.302PAGE PREVIEW . . . . .	112

---

---

1.303FILES . . . . .	112
1.304ESC-SEQUENCES . . . . .	113
1.305CHARACTER HEIGHT . . . . .	113
1.306LANDSCAPE . . . . .	113
1.307BOOK . . . . .	113
1.308COVER . . . . .	114
1.309SINGLE/DOUBLESIDED . . . . .	114
1.310PERFORATE . . . . .	114
1.311OUTPUT DEVICE . . . . .	114
1.312SHELL INTERFACE . . . . .	115
1.313AREXX PORT HISPEED . . . . .	115
1.314ARexx commands HiSpeed . . . . .	115
1.315MACROS . . . . .	117
1.316FILTER . . . . .	117
1.317SLEEP-ICON . . . . .	117
1.318RECOVER . . . . .	118
1.319REGISTER GOLDED . . . . .	119
1.320UPDATES . . . . .	120
1.321Upgrading to GoldED 4 . . . . .	120
1.322CREDITS . . . . .	121
1.323ADDRESS . . . . .	121
1.324 GoldED . . . . .	122

---

# Chapter 1

## GED.HYPER

### 1.1 main

©1996 Dietmar Eilert. All Rights Reserved.

#### Topics

Features	Command list
Licence	Variables
Introduction	Tools
Required system	Register GoldED
Getting started	Updates
Mouse configuration	Upgrading to GoldED 4
Menu	Credits
Keyboard	Address
ARexx port	Index

#### Menus

Project menu	View menu
Block menu	Extras menu
Layout menu	Macro menu
Search menu	About GoldED

### 1.2 FEATURES

#### FEATURES

Folding	Nested folding
Undo & redo	Unlimited configurable undo & redo
Menu configuration	Configurable menus
Options local	Notebook windows simplify configuration
ARexx port	ARexx port offers 250+ commands and options
Filetypes	Local configuration, filetype detection
Toolbar	Configurable toolbar
Mouse configuration	Configurable mouse buttons
Drag & Drop	Move text using the mouse

Syntax highlighting	Configurable syntax highlighting
QuickReference	QuickReference capability for includes, guides, ...
APC	Automatic phrase completion based on dictionary
Shifting	Indentation of blocks
HotKey	HotKey support, commodity support
File list	Open files are listed in the menu
Hide	Hide windows
WordWrap	Formatting functions, word wrap
SmartIndentation	Context-sensitive indentation (SmartIndentation)
AutoBackup	Automatic backups
MenuHelp	Menu help based on AmigaGuide library
Language	Localized
Debugger	Breakpoints display: source level debugger frontend
Printer	Printer configuration available
Spooler	Asynchronous printing
Multiselection	File requesters support multiselection
XPK support	XPK support: (de)crunching & encryption
Search/ASCII table	ASCII character selection table
Display mode	Configurable GUI: display mode, fonts
Clipboard	Clipboard support
Project/Open fast	Fast loading
AutoCase	Automatic online case correction
Parenthesis check	Automatic online parenthesis check
QuickFunc	Displays a table of functions defined in source code
Insertion of columns	Insertion of columns, removal of columns
Dynamic tabs	Fixed, regular, solid and dynamic tabs
Character set remap	Character set remap (e.g. Amiga to MS-DOS)
Quickstarter	Quickstarter
Sequences	Sequence recording
Templates	Templates
Global search	Global search handles multiple files
Search/Replace History	Search & Replace history
File hunter	Built-in file search
Configurable gadgets	Configurable gadgets
Preview	Preview display mode
Wildcards	Search functions support wildcards

and many more ...

## 1.3 LICENCE

### LICENCE

This licence agreement affects all programs, manuals and disks of the editor package GoldED, release 4 or later. You agree to accept this licence. Violations of any of the topics below will terminate your rights to use or otherwise handle (e.g. distribute) the package.

You may not duplicate, distribute, hire or sell a registered GoldED package or GoldED keyfiles. Keyfiles are ©1995-1996 Dietmar Eilert. You may not distribute original GoldED keyfiles nor are you allowed to create, use or distribute reverse-engineered keyfiles. Distribution of keyfiles is a violation of copyright laws and will be prosecuted by the copyright owner(s).

The unregistered trial version version may be made available for downloading

by online services - e.g. FTP sites - if the package is made available free of charge. No other forms of distribution are allowed unless you have written permission of the copyright owners, including, but not limited to, CD ROM distribution or distribution on cover disks. A distribution licence for GoldED can be purchased from the copyright owner(s).

The copyright owner(s) reserve the right to interdict distribution at any time if a distributor fails to obey to this licence.

You may not modify the package: you may not change the compilation, you may not modify, reverse-engineer or translate files and you may not add or remove files. You are not allowed to distribute translations of material found in this package unless authorized by the copyright owner(s). All translation rights reserved.

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 be liable 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

This editor has been designed as a powerful and flexible tool made for software developers: a fast and reliable development environment. Installation and usage have been designed to be simple and clear, the user interface is functional and unobtrusive.

The internal design is event based: the editor waits for events like a keystroke or a menu selection and then calls a dispatcher to perform the appropriate action. Action is not hard coded: you may assign any of the editor's internal functions to events (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 "don't panic" 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 function calls only, performing some straight-forward kind of action. The editor relies on ARexx as far as conditional statements, variables and flow control are concerned.

## 1.5 REQUIRED SYSTEM

### REQUIRED SYSTEM

Minimum requirements are OS2.04, 68000 and 1 MB free RAM. However, this

---

configuration isn't suitable for working effectively. Besides, GoldED has not been tested with a 68000. Suggested minimum system for average performance is a 68020 CPU and 2 MB free RAM. Total memory requirements including memory consumed by a compiler and the OS will be 8 MB RAM to 16 MB RAM. Usage of syntax highlighting increases resources requirements: a fast CPU and a fast graphics adapter should be installed. Several features (localized requesters, cursor control in listviews, free color design of menus, multiple selection in listviews, etc.) require OS3 and are not available if OS2.x is installed. We strongly recommend to install/rekick OS3; GoldED's performance increases noticeably if OS3 is available.

Additional software requirements - the following packages should be installed on your machine (available at most Amiga freeware & PD distributors):

- o Reqtools library (required)
- o XPK library (optional)
- o AmigaGuide library (optional)

## 1.6 GETTING STARTED

### GETTING STARTED

You may start this editor the way other programs are started, too: doubleclick at the icon or type the name ("golded:golded") in a shell window. However, these methods are not recommended to start GoldED: the main disadvantage is that a new instance of GoldED is started every time, consuming memory and reducing usability. We recommend that you use the QuickStarter instead:

#### QuickStarter

The quickstarter ED is a small (4 KB) GoldED frontend. You can use it as if it were a real editor. For example, you could type "ed letter" to edit the file "letter". The advantage of the quick starter is its ability to pass a new job to a running instance of GoldED (if there is one - a new GoldED process is started if necessary). The quick starter may be made resident (the main editor may not be made resident). The source code of ED is available in the GoldED drawer (developer/examples/quickstarter).

#### Arguments

The quickstarter and GoldED support several startup options (besides a list of file names to be edited) explained below:

HIDE (example: ED HIDE)

Use the HIDE option but don't supply file names if you want the editor to stay in the background initially (waiting for Hotkey activation and ARexx commands). The editor will not open a window.

FILETYPE (example: ED sys:letter FILETYPE=TEXT)

Overrides automatic filetype detection: the example above tells GoldED to provide the environment usually used for files of the type "TEXT". You'll

---

learn more about file types and local configuration later in this manual. Usually configuration is controlled by automatic file type detection based on file names ("sys:letter" in the example above). The FILETYPE option overrides automatic file type detection: configuration selection now depends on the symbolic file name given as argument (e.g. "TEXT", ".c", ".guide"). See options global for more information on file types.

STICKY (supported by the quickstarter only)

The quickstarter supports synchronous and asynchronous operation (unless the HIDE option is used to run the editor in the background) depending on whether or not the STICKY option is used. A call to ED returns immediately if STICKY is not specified. The Quickstarter returns after the window opened by ED is closed if the STICKY option is specified (GoldED itself may keep on running if there are further windows to handle): ED returns when the user quits out of the file. This option is designed to be used to synchronize GoldED with a program running the editor to edit a file. Typical usage: GoldED launched by a browser to edit e-mails. Example:

```
ED mail:answer STICKY
```

AREXX/K

This option sets the name of GoldED's ARExx port if the editor is not yet running (it is ignored if GoldED is running already). Defaults to "GOLDED.1". Example:

```
ED AREXX="PORT.1".
```

SESSION/K

The SESSION option can be used to restore the environment of a previous session quickly (saved using the macros/session save menu). Example:

```
ED SESSION="golded:sessions/old.session".
```

MenuHelp

The editor's help facility is based on Commodore's AmigaGuide tool and the AmigaGuide library. Doubleclick the manual icon to make AmigaGuide display the help file. You may use MultiView of OS3.0 to read the guide file, too. GoldED supports menu help: press the <HELP> key while selecting a menu to get explanations related to the item you selected. Help windows block text windows if your machine doesn't run OS3.

Language

Locale library of OS2.1 or better is supported as far as requesters are concerned: Requesters will use German translations if you have selected German to be your default language (using the 'locale' preferences of your Workbench). English, French, German, Italian and Swedish are available so far; further languages may be available in the future. Locale settings do not affect menus: menus are not part of the editor itself but external files. The default menu definition file is available in several languages. It is installed during GoldED installation.

---

GUI (Graphical User Interface)

The editor's user interface supports the look & feel of OS3. Most gadgets support keyboard activation: underlined characters within gadget labels indicate shortcuts. Shortcut keys allow you to move a slider gadget or to activate a button gadget without your mouse. Use the shift key simultaneously to toggle the direction of changes - e.g. to move a slider gadget one step to the left instead of one step to the right. Use the Amiga key simultaneously to activate another gadget from within a string gadget (your input would go to the string gadget if R-Amiga is not used). Amiga-X clears string gadgets. CTRL+L inserts a form feed code, CTRL-ESC an ESC code, CTRL-TAB a tab code.

## 1.7 MOUSE CONFIGURATION

### MOUSE CONFIGURATION

A simple mouse click into a text window positions the cursor. Doubleclick into the window to mark the word under the cursor. Hold down the mouse button and drag the mouse pointer over the text to mark lines or single words. This editor offers two marker resolutions: character resolution and line resolution. Hold down the SHIFT key while marking to change from character resolution (useful to mark words) to line resolution (useful to mark paragraphs).

Drag & Drop

Clicking at a marked section while holding down the ALT key activates the drag & drop mode: the cursor shape changes to a symbolized block, indicating that you can now move the block using the mouse. Release the mouse button to have the text moved to the mouse position.

All mouse functions described above can be customized; this is a description of factory defaults only.

## 1.8 MENU

### MENU

Project menu	View menu
Block menu	Extras menu
Layout menu	Macro menu
Search menu	

GoldED offers almost unlimited user configuration - there is no "standard" appearance. Colors, resolution, fonts, menus: all major parts of GoldED are configurable. This text describes the default configuration shipped with the editor. You'll probably want to customize the standard setup using extras/options local or extras/options global .

---

## 1.9 PROJECT MENU

PROJECT MENU  
Menu tree of project menu

project/clear window	project/save and close
project/open	project/print
project/open fast	project/name
project/open new	project/information
project/open original	project/protection
project/insert	project/current dir...
project/append	project/current dir
project/save	project/quit all & exit
project/save as	project/quit buffer
project/save as XPK	

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, print).

### 1.10 project/clear window

project/clear window of: PROJECT MENU

Clears contents of current window. Protection bits are set to the defaults (see extras/options local/misc ). The path is reset to the current path, the file name is reset to "unnamed". You are asked for confirmation if the text has been modified.

### 1.11 project/open

project/open of: PROJECT MENU

Loads a new text (file requester selection). Discards current text if any was loaded. A confirmation request has to be answered if the text has been modified.

Multiselection

File requesters offered by GoldED usually support multi selection: you may select more than one file (hold down the SHIFT key while selecting files). Multi selection handling depends on the context: project/append loads all selected files into one window, project/open opens a new window for each file.

AppWindows

Text windows are AppWindows if GoldED is running on the Workbench screen: drag icons of text files over a text window to have those files loaded by GoldED. The files are appended to existing text. Multi selection is

supported: You may drag more than one file over a window using extended selection (hold down the shift key while you select icons).

XPK files (see project/save as XPK ) are recognized and decompressed if the XPK libraries are available. This function recognizes tabs (dez. 9): Tabs are replaced by spaces (dez. 32) while loading. Usually one tab code is expanded to eight spaces. Extras/options global/misc allows you to customize substitution. This function uses a small input buffer to speed up operation. Better performance (at the cost of higher RAM usage) is provided by project/open fast .

Files usually are scanned for folded sections after loading if the AutoFold feature has been enabled: marked sections are folded before the text is displayed for the first time. The AutoFold feature can be turned off ( extras/options local/misc ) in order to decrease loading time.

All file related functions of GoldED can handle text files only. Don't 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).

## 1.12 project/open fast

project/open fast of: PROJECT MENU

FastLoad

A fast replacement for Project/Open . This function uses a large cache (same size as the file you want to load) in order to increase loading speed. Performance advantage (compared to Project/Open ): up to 50% faster. RAM requirements example: about 900 K are used if loading a 400K file: 400K cache and 500K to store the file in the internal format. The cache is freed immediately after the file has been loaded.

Technical information: Exact RAM usage depends on the average line width. 28 Bytes are allocated for each text line (to store general information) plus sufficient RAM to store the line's text. That's why a 40K text consisting of empty lines only consumes much more RAM than a normal 40K text.

This function doesn't recognize tab codes: tab codes usually are not expanded to spaces. That's not a problem if loading documents generated by GoldED: files saved by GoldED don't include tabs. Ignoring tabs allows usage of a faster loader. You should use the normal load function ( Project/Open ) if loading files generated by other editors. Tabs found in those files are displayed as inverted "T" otherwise. The editor attempts to reduce this problem using an automatic detection routine: Fallback to normal loading if a tab code is found in the first 1500 bytes of the document.

## 1.13 project/open new

---

project/open new of: PROJECT MENU

Loads a document into a new window (see project/open ).

## 1.14 project/open original

project/open original of: PROJECT MENU

Reloads current file from disk. Useful after you have made some changes but want to switch back to the original. This function uses slow loading (see project/Open fast ), i.e. tabs are substituted by spaces.

## 1.15 project/insert

project/insert of: PROJECT MENU

Inserts a file at current cursor position (before the 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 "multiselection").

## 1.16 project/append

project/append of: PROJECT MENU

Appends 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. Quite useful to join a couple of files together.

## 1.17 project/save

project/save of: PROJECT MENU

Saves the text to the file displayed in the window's title bar. Old copies are overwritten without asking for confirmation if overwrite mode ( extras/options local/misc ) has been enabled. Turn on backup creation ( extras/options local/misc ) if you want the editor to backup the old version. Automatic backups (e.g. every ten minutes) are available, too: the backup path is set using the extras/options local/misc requester, the backup interval is set by the extras/options global/misc requester. Backups are written to the directory a text has been loaded from if no dedicated backup path is selected. A new suffix ".bak" is appended to all backup files.

---

This function is not available if a window is read-only to prevent you from accidentally overwriting important files. QuickRef windows are read-only (see QuickReference ). Project/protection allows you to change the windows status.

## 1.18 project/save as

project/save as of: PROJECT MENU

Saves current buffer (compare project/save ) to the file you select in a file requester (defaults to the name displayed in the window title). This function is not available if the buffer status has been set to read-only (see project/save ).

## 1.19 project/save as XPK

project/save as XPK of: PROJECT MENU

XPK support

Saves current file in compressed mode. See extras/options local/misc on how to select a compression mode. Some compression modes require a password - the password is set using the extras/options global/misc configuration. Encrypted files can not be loaded if the password isn't set correctly (neither by GoldED nor by other editors).

This function requires installed XPK libraries. XPK libraries are not shipped with GoldED. Other programs may or may not be able to handle XPK files. Compilers usually can not load XPK files.

## 1.20 project/save and close

project/save and close of: PROJECT MENU

Saves a buffer and closes its window. The file is not saved if the window status has been set to read-only (see project/save ) or if the file could not be saved successfully. GoldED is either terminated or kept running in the background after the last window has been saved depending on the "resident" setting ( extras/options global/misc ).

## 1.21 project/print

project/print of: PROJECT MENU

Sends current text to the printer. Several output-related options can be set using the `extras/options local/printer` configuration. Further configuration depends on the Workbench printer setup (`prefs/printer`).

LaserJet/DeskJet owners may use the HiSpeed printing utility `extras/HiSpeed` instead - it provides better configuration options (e.g. book printing mode). HiSpeed is a shareware program; a HiSpeed registration is part of every GoldED Pro/NET registration.

## 1.22 project/name

project/name

Allows you to change the name of the current buffer (you are prompted for a new name). The buffer is renamed, however, the file the text has been loaded from is not renamed (`extras/files` allows you to rename files). Changing the name triggers file-type-detection (`extras/options global/filetypes` and may result in a modified environment

## 1.23 project/information

project/information of: PROJECT MENU

Presents some statistics related to the current text: Bytes, lines, folds and non-ASCII-characters (codes above 127) are counted. The width of the longest line is determined. Invisible linefeed characters are included in the figures. Additionally, a global figure is displayed: total UNDO RAM consumption. This figure should help you to customize the undo buffer size (`extras/options global/misc`).

## 1.24 project/protection

project/protection of: PROJECT MENU

Displays protection bits of the current text. Please refer to your AmigaDOS manual if you are unfamiliar with the meaning of these bits. Usually only the script bit should be changed: Enable the script bit if creating batch file. Changes won't have any effect until you save the text. These bits are set to a default state by `project/clear window` (`extras/options local/misc`).

---

## 1.25 project/current dir...

project/current dir... of: PROJECT MENU

A filerequester allows you to select a new "current path". The current path defaults to the current path of the application running GoldED (e.g. current path of the shell window GoldED has been started from).

The current path is used by several functions of GoldED (e.g. displayed as default path by project/open new ). It is passed as current path to programs run by GoldED, too. A few functions (e.g. project/open ) ignore the current path: those functions extract path information from the document's name. Setting the current path doesn't change the text name. Project/Clear window resets the path to the current directory.

## 1.26 project/current dir

project/current dir of: PROJECT MENU

Instructs the editor to use the document's path as default "current path" (see Project/Current dir... ).

## 1.27 project/quit all & exit

project/quit all & exit of: PROJECT MENU

Closes all text buffers and terminates GoldED. A confirmation request is displayed if one of more buffers have been modified. The editor is completely removed from memory and is not kept running in the background (see Project/Quit buffer ) unless a debugger has locked GoldED: GoldED can not be terminated during a debugger session.

## 1.28 project/quit buffer

project/quit buffer of: PROJECT MENU

Closes current window. A confirmation request is displayed if the document has been modified. GoldED is either terminated or kept resident in the background (waiting for a hotkey ) after the last window has been closed. extras/options global/misc allows you to choose the behaviour you prefer (resident switch).

GoldED internally uses asynchronous memory management: you won't have to wait until all memory has been freed after you close a window (this job is handled by a background task). Your Amiga might feel slightly slower than usual while the background task is busy - especially if the text buffer has been large

---

(200 KB or more).

## HotKey

GoldED is not removed from memory after the last window has been closed if hotkey support has been activated ( extras/options global/misc : resident switch enabled). Instead, the editor will wait for hotkey activation (ALT + right SHIFT + RETURN) in the background and then reopen its windows. Other ways and means to reopen a GoldED task waiting in the background include the Commodities Exchange program of your Workbench and macro execution (commands sent to the ARexx port of GoldED will wake GoldED up). Main advantage of keeping the editor resident is reduced startup time. Main disadvantage is increased memory consumption.

## 1.29 BLOCK MENU

### BLOCK MENU Menu tree of block menu

block/mark	block/delete
block/mark line	block/move
block/mark paragraph	block/column
block/mark hide	block/block sort
block/clipboard cut	block/block to uppercase
block/clipboard copy	block/block to lowercase
block/clipboard paste	block/block print
block/paste column	block/block save
block/copy	

All functions of the block menu are related to the management of "blocks", which are marked sections of lines. Usually one would use menu functions or corresponding keyboard shortcuts to mark lines, however the mouse may be used, too: 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.

## 1.30 block/mark

block/mark of: BLOCK MENU

Marks a block. This function is used to mark single characters as opposed to block/mark line. Cursor movements resize the block. Further usage of this menu toggles flow marker mode on or off. Use block/mark hide to turn the marker off completely. Cursor movements either set the block start or the block end depending on the cursor position (the block start is set if the cursor is closer to the block start than to the block end).

Some functions of this editor require line blocks. For example you can't mark a single word and block-format it. GoldED automatically converts blocks to line blocks if required.

### 1.31 block/mark line

block/mark line of: BLOCK MENU

Marks beginning or end of a block. This command is used to mark whole lines (see block/mark ). Sets block start if this function is used for the first time (no marked lines so far). Sets block end if you use this function a second time. All lines between start and end position are marked. Handling is different if you use this function after a block has been selected already: The block end is set if the cursor position is closer to the end of the current block than to its start. Otherwise the start position is set to the cursor's line. Use block/mark hide to turn the marker off.

### 1.32 block/mark paragraph

block/mark paragraph of: BLOCK MENU

Marks paragraph: all lines between the previous empty line and the next empty line are highlighted. Use this function again (or Block/Mark hide ) to remove the selection.

### 1.33 block/mark hide

block/mark hide of: BLOCK MENU

Removes block selection set using block/mark (or other marker functions).

### 1.34 block/clipboard cut

block/clipboard cut of: BLOCK MENU

Cut selected parts (see block/mark / mouse ) from text. These lines are written to the clipboard, ready to be inserted into applications supporting the clipboard device (e.g. into a shell window by pressing AMIGA & V or into a GoldED document using block/clipboard paste ). The clipboard offers several storages called "units" and thus is able to keep several text blocks. Standard unit (used to exchange data between applications) is unit 0. You shouldn't use this function to move/copy data within GoldED documents: block/copy or block/move perform the same task much faster.

### 1.35 block/clipboard copy

block/clipboard copy of: BLOCK MENU

Writes selected text (see block/mark , mouse ) to the clipboard device, ready to be inserted into applications supporting this device. The clipboard offers several storages called "units" and thus is able to keep many blocks of data; standard unit (used to exchange data between applications) is unit 0. This function doesn't affect your current text, the selection is not deleted (see block/clipboard cut ). Use block/clipboard paste to insert text posted to the clipboard into a GoldED document.

### 1.36 block/clipboard paste

block/clipboard paste of: BLOCK MENU

Insert clipboard contents (if any are available) into current text. An empty clipboard is reported as "clipboard error". Single words found in the clipboard are inserted at the current cursor position. Paragraphs (i.e. multiple lines) found in the clipboard are treated differently: they are inserted before the current line.

Clipboard

The clipboard (actually the 'CLIPS:'-directory) is used by many applications to exchange data. It offers several storages called "units" and thus is able to keep several blocks of data simultaneously. GoldED can access any unit (see CLIP command), though usually only unit 0 is used. Postings to the clipboard are IFF files. GoldED supports IFF/FTXT clipboard access (text files). Other files posted to the clipboard can not be insert into editor windows. Example usage: mark a text in a shell window, press AMIGA + C to copy these lines to the clipboard unit 0 and then reinsert them into a GoldED window using AMIGA + V. A paste operation won't remove the data from the clipboard, i.e. you can paste the same data multiple times.

### 1.37 block/paste column

block/paste column of: BLOCK MENU

Inserts clipboard contents "vertically" at current cursor position: the clipboard contents are mixed to the existing lines. Useful to paste a second column of text next to an existing column of text. Depending on insertion mode ( layout/insert mode ) pasted text is either inserted into the existing text or overwrites existing text. While insert mode is recommended for pasting single words, overwrite mode should be used to create multi-column documents. You will get a 'clipboard error' if the clipboard is empty. Use block/clipboard copy to move text sections to the clipboard.

### 1.38 block/copy

block/copy of: BLOCK MENU

Copies selected text (see block/mark , mouse ) to current cursor position. Useful to duplicate sections within editor windows. Can not be used to copy text from one window to another window. Use block/clipboard cut and block/clipboard paste to exchange data between windows or applications.

### 1.39 block/delete

block/delete of: BLOCK MENU

Deletes highlighted section (see block/mark or mouse ).

### 1.40 block/move

block/move of: BLOCK MENU

Moves highlighted text (see block/mark , mouse ) to current cursor position (if single words have been marked) or to the cursor line (if lines have been marked). Can not be used to move text from one window to another window. Use block/clipboard cut / block/clipboard paste to exchange data between windows or applications.

### 1.41 block/column

block/column of: BLOCK MENU

This menu offers column-oriented editing functions:

- block/column delete
- block/column insert
- block/column text
- block/append text

### 1.42 block/column delete

block/column delete of: BLOCK MENU

Removal of columns

Deletes cursor column from selected lines (see block/mark on how to select

---

lines). Move the cursor to the column to be removed before you call this function (e.g. move cursor to column 10 if you want to delete this column from all block lines). You shouldn't use this function to change indentation of a paragraph to avoid accidentally deleting text; use `layout/indent block` instead.

### 1.43 block/column insert

block/column insert of: BLOCK MENU

Insertion of columns

Inserts a new column into highlighted lines (see `block/mark` on how to mark lines). Move the cursor to the correct position before you call this function. Example: Move cursor to column 40 if you want to insert one space character at position 40 into all block lines.

### 1.44 block/column text

block/column text of: BLOCK MENU

Allows you to insert a text at any position into multiple lines simultaneously. Example usage: Mark some lines (see `block/mark`), move cursor to the destination column (e.g. column 1) and call this function. A requester pops up, asking you for a text to be inserted. Enter 'Prototype'. GoldED inserts this text into all block lines at column 1. This function simplifies table creation (insertion of '|' creates a vertical line).

### 1.45 block/append text

block/append text of: BLOCK MENU

Allows you to append text at the end of all highlighted lines (see `block/mark` on how to select lines). Example usage: Mark some lines and call this function. A requester pops up, asking you for the text to be appended to the lines. A semicolon is added at the end of each line if you enter ";".

### 1.46 block/block sort

block/block sort of: BLOCK MENU

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

---

## 1.47 block/block to uppercase

block/block to uppercase of: BLOCK MENU

Converts selected text (see block/mark ) to uppercase characters. This function uses the locale library if available to convert non-ASCII characters (e.g. 'ß') correctly. The locale library is part of OS2.1 and OS3.0 or better. It is not available is using OS2.04.

## 1.48 block/block to lowercase

block/block to lowercase of: BLOCK MENU

Converts selected text (see block/mark ) to lowercase characters. This function uses the locale library if available to convert non-ASCII characters (e.g. 'ß') correctly. The locale library is part of OS2.1 and OS3.0 or better. It is not available is using OS2.04.

## 1.49 block/block save

block/block save of: BLOCK MENU

Saves selected lines (see block/mark ) to disk (see project/save . You are asked for a file name.

## 1.50 block/block print

block/block print of: BLOCK MENU

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

## 1.51 LAYOUT MENU

LAYOUT MENU  
menu tree of layout menu

layout/set right margin	layout/block right
layout/use current margin	layout/block centered
layout/insert mode	layout/indent

layout/solid tabs	layout/paragraph justified
layout/word wrap	layout/paragraph left
layout/templates	layout/paragraph right
layout/AutoCase	layout/paragraph centered
layout/block justified	layout/NumPad = movement
layout/block left	

Formatter

All functions of the layout menu are related to formatting a text and layout modifications.

Paragraph vs. block

Some formatting functions affect selected lines only (refer to block/mark if you don't yet know how to mark lines). Others affect the 'current paragraph' of a text: all lines between the previous and the next empty line.

## 1.52 layout/set right margin

layout/set right margin of: LAYOUT MENU

Sets right margin to match current cursor position. Example usage: move cursor to column 80, then call this menu. This will set the right margin to column 80. Margin values are considered by formatting operations only (e.g. WordWrap). Layout margins don't necessarily restrict cursor movements. This function allows you to adjust the right margin faster than using the layout configuration ( extras/options local/layout ).

## 1.53 layout/use current margin

layout/use current margin of: LAYOUT MENU

Formatting functions (including word wrap) ignore the left border settings ( extras/options local/layout ) if this menu appears checked: the left margin is set to the indentation of the cursor line instead. Usage of this option simplifies reformatting a text using a complex indentation scheme.

## 1.54 layout/insert mode

layout/insert mode of: LAYOUT MENU

Toggles writing mode from insert mode to overwrite mode 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: INSR (insert) or OVER (overwrite).

---

## 1.55 layout/solid tabs

layout/solid tabs of: LAYOUT MENU

Toggles tab mode from solid tabs to light tabs and vice versa. Solid tabs insert blank characters into your text. Depending on whether insert mode is on or off (see layout/insert mode ) the existing text is either indented or overwritten. Light tabs do not overwrite text - the cursor is simply moved to the next tab position. See extras/options local/tabs on how to set tab positions.

## 1.56 layout/word wrap

layout/word wrap of: LAYOUT MENU

WordWrap

Toggles word wrap mode on/off. Current status is displayed in the screen's title bar (WRAP). Word wrap enabled: the editor automatically inserts a return if the cursor is moved beyond the right margin while typing. Words exceeding the right margin are moved to the next line. Additionally, the paragraph is reformatted if automatic reformatting has been enabled ( extras/options local/layout ). See extras/options local/layout on how to adjust the right margin. Word wrap is very useful if you edit a plain ASCII text. Word wrap should be disabled if editing source files or tables.

## 1.57 layout/templates

layout/templates of: LAYOUT MENU

Templates

Toggles template mode on/off. Current status is displayed in screen's title bar (TMPL). If template mode is on, the editor will look for search patterns (templates) during user input. If a template is found, template-specific action as set up by the user is performed. Using templates you could make the editor replace "SNC" by "sincerely" immediately while you are typing. Several action types are available, including playback of recorded sequences , internal commands or execution of ARexx macros. Template setup is described in the layout/templates section.

## 1.58 layout/AutoCase

layout/AutoCase of: LAYOUT MENU

Toggles AutoCase mode on/off (automatic uppercase/lowercase correction).

---

AutoCase correction is based on a dictionary ( extras/options local/words ).

## 1.59 layout/block justified

layout/block justified of: LAYOUT MENU

Reformats marked lines (see block/mark ) to make them 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: don't attempt to format tables. Use extras/options local/layout to adjust layout margins.

## 1.60 layout/block left

layout/block left of: LAYOUT MENU

Reformats marked lines (see block/mark ) to make them left aligned. Empty lines are not removed during formatting, i.e. your document's basic structure remains unchanged. Multiple spaces are removed: don't attempt to format tables. Use extras/options local/layout to adjust layout margins.

## 1.61 layout/block right

layout/block right of: LAYOUT MENU

Reformats marked lines to make them right aligned. Empty lines are not removed during formatting, i.e. your document's basic structure of paragraphs remains unchanged. Multiple spaces are removed: don't attempt to format tables. Use extras/options local/layout to adjust layout margins.

## 1.62 layout/block centered

layout/block centered of: LAYOUT MENU

Reformats marked lines to make them appear centered within the layout area. Empty lines are not removed during formatting, i.e. your document's basic structure of paragraphs remains unchanged. Multiple spaces are removed: don't attempt to format tables. Use extras/options local/layout to adjust layout margins.

---

## 1.63 layout/indent

layout/indent of: LAYOUT MENU

This menu offers indentation-oriented editing functions:

```
layout/indent block
layout/line: left
layout/line: right
```

## 1.64 layout/indent block

layout/indent block of: LAYOUT MENU

Shifting

Changes indentation of marked lines ( block/mark ). Use the arrow gadgets to shift the block to the left or to the right. Currently selected keyboard-tab distance (see extras/options local/tabs ) is used as default indentation step. Indention functions are assigned to the cursor keys, too (see cursor keys ).

## 1.65 layout/indent line

layout/indent line of: LAYOUT MENU

```
layout/line: left
layout/line: right
```

Modifies indentation of the cursor line. Shifting distance is the tab distance set using extras/options local/tabs .

## 1.66 layout/paragraph justified

layout/paragraph justified of: LAYOUT MENU

Reformats current paragraph (see Paragraph vs. block ) in order to make the lines left & right aligned. Please refer to layout/block justified for further explanations.

## 1.67 layout/paragraph left

layout/paragraph left of: LAYOUT MENU

Reformats current paragraph (see Paragraph vs. block ) in order to make the lines left aligned. Please refer to layout/block left for further explanations.

## 1.68 layout/paragraph right

layout/paragraph right of: LAYOUT MENU

Reformats current paragraph (see Paragraph vs. block ) in order to make the lines right aligned. Please refer to layout/block left for further explanations.

## 1.69 layout/paragraph centered

layout/paragraph centered of: LAYOUT MENU

Reformats current paragraph (see Paragraph vs. block ) in order to center the lines within the layout area. Please refer to layout/block centered for further explanations.

## 1.70 layout/NumPad = movement

layout/NumPad = movement of: LAYOUT MENU

Enables/disables special configuration of NumPad keys. Standard assignments are used if this option is disabled (numbers). PC-bindings (e.g. PgUp = previous page) are used if "NumPad = movement" is enabled.

## 1.71 SEARCH MENU

SEARCH MENU  
menu tree of search menu

search/find	search/reference...
search/find next	search/complete
search/find previous	search/ASCII table
search/replace	search/ASCII code ?
search/replace next	search/ASCII code insert
search/count pattern	search/character set

search/check	search/function list
search/matching bracket	search/function find
search/reference	

## 1.72 search/find

search/find of: SEARCH MENU

Shows a requester to enter the text to search for. You may choose the search to be case sensitive or not by setting the appropriate checkmark gadget in the requester. The OK gadget makes the editor search the next (i.e. after current cursor position) occurrence of the text in the current document. Use the FIRST gadget to find the first occurrence of the search pattern in the document. extras/project filter allows you to search multiple files.

Wildcards

GoldED supports AmigaDOS wildcard pattern matching if <wildcards> are enabled, thus giving you access to advanced search functions: The reserved pattern matching characters (e.g. "\*" or "|") are interpreted in wildcard mode instead of being searched literally. Valid patterns are described in your AmigaDOS manuals. Wildcard search is based on lines. The pattern "Prototype\*" would make the editor look for a line beginning with "Prototype". Add a leading "\*" and a trailing "\*" if looking for words within lines. The editor automatically switches to fast non-wildcard search if the search string doesn't contain any wildcard characters.

Search/replace history

Use the block gadget next to the arrow gadget to search for the marked text. Use the arrow gadget to open a search/replace history listview. The listview offers previously used search/replace strings as well as the word under the cursor for fast selection.

## 1.73 search/find next

search/find next of: SEARCH MENU

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

## 1.74 search/find previous

search/find previous of: SEARCH MENU

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

---

of the pattern to search for. Use `search/find` to adjust the search pattern.

## 1.75 search/replace

search/replace of: SEARCH MENU

Displays 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`). Please refer to the `search/find` section for further details (history, wildcards).

## 1.76 search/replace next

search/replace next of: SEARCH MENU

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

## 1.77 search/count pattern

search/count pattern of: SEARCH MENU

Displays a requester to enter a search text. You may choose the search operation to be case sensitive by using the appropriate checkmark gadget. Select the <FIRST> gadget to count all occurrences of the search pattern within your text or the <OK> gadget to start counting at the current cursor position. Please refer to the `search/find` paragraph for further details (history, wildcards).

## 1.78 search/check

search/check of: SEARCH MENU

Checks for correct use of brackets '(' within current line. You will be warned if there are more opening braces than closing ones or vice versa. Brackets in strings are recognized and ignored.

---

## 1.79 search/matching bracket

search/matching bracket of: SEARCH MENU

Moves cursor to matching bracket. Handles several bracket types: (), , etc. Useful to check levels of execution within a program. Example usage: move cursor over first opening parenthesis "{" of a function (C source code), then use this function twice: the cursor position shouldn't change if nesting of brackets is correct.

## 1.80 search/reference

search/reference of: SEARCH MENU

QuickReference

Opens a help text related to the word your cursor is placed over. Example usage: Type 'struct RastPort', move the cursor over 'RastPort' and call this function. A new text window should pop up, showing you the file 'graphics.h' from your compiler's include directory. The cursor will be placed in the first line of the structure definition 'RastPort'. This feature requires that the reference system has been set up correctly. Files displayed by the reference function are read-only to prevent you from accidentally overwriting those files.

Setting up the reference system

You'll have to set up GoldED's reference system first in order to have this command work properly (i.e. tell the editor where to look for files like 'graphics.h'): Use extras/options local/reference to do this. Setting up the reference system basically consists of selecting the files or directories to be referenced (where to look for help). C-sources, include files, AutoDocs, BASIC programs, assembler sources and Pascal sources are understood by the editor. Add those files to the list of files to be scanned. A database is created during setup of the reference system based on these files. The database contains a list of keywords and information where to look for help related to the keywords. The database is created automatically: the editor scans all files and directories you have named. It recognized the text types of those files (checking the file name suffix) and selects a matching parser to extract keywords. Recognized file name extensions can be modified ( search/function list ). Adding further parser (same requester) dealing with so-far-unknown text types allows you to extend the editors reference capabilities. search/function list explains how to program parsers.

## 1.81 search/reference...

search/reference... of: SEARCH MENU

Prompts you for a string to be referenced: The editor then attempts to display a help text related to this string. Similar to search/reference except that the word to be checked is entered in a string requester (as

---

opposed to referencing the word under the cursor). Example usage: Enter "RastPort" (not "struct rastPort") to have the structure definition "RastPort" displayed. This feature will not work until the reference system has been set up correctly search/reference .

## 1.82 search/complete

search/complete of: SEARCH MENU

COMPLETE DICTIONARY

APC

Completes the word your cursor is placed over depending on the dictionary currently loaded (see IntelliSense ). Example usage: type 'swin', then call this function. 'swin' is replaced by 'struct Window' if the C-dictionary is present (see extras/options local/words ). A smart algorithm is used to find the full form in the dictionary: "swindow' or "swdow" would have been recognized, too (however, the first letter of the word to be completed must match the first letter of the full form). The larger your dictionary grows the more detailed your abbreviations will have to be to ensure unique identification. We recommend to keep the dictionary small and to add often-used expressions only.

COMPLETE CONTEXT

Completes the expression under the cursor depending on the context of the document ( IntelliSense ). Example usage: Place the cursor over "al" and press SHIFT-SPACE: "al" is expanded to an expression starting with "al" found in the same document.

## 1.83 search/ASCII table

search/ASCII table of: SEARCH MENU

Opens a character selection requester: All characters of your current font are listed in a table. Pick the character you are looking for: it is inserted at the current cursor position. Useful to insert special characters if you don't know the character's keystroke sequence.

## 1.84 search/ASCII code ?

search/ASCII code ? of: SEARCH MENU

Shows ASCII code of character under cursor.

---

## 1.85 search/ASCII code insert

search/ASCII code insert of: SEARCH MENU

ESC, FF, TAB

Allows you to insert special codes into the text (escape, form feed, tab).  
Intended usage: preparing a text for printing.

INSERT ASCII CODE

Prompts you for an ASCII code to be inserted at the cursor position. Available codes (ASCII tables) usually can be found in printer manuals. This function allows you to insert control codes into a text. For example, the sequence 27 91 49 109 is recognized as "bold printing" command by many printers. Most text viewers support control codes, too, setting color and style of the displayed text.

## 1.86 search/character set

search/character set of: SEARCH MENU

Character set remap

Allows you to select a character set conversion file to be applied on the text: the text is remapped according to the conversion table. Remapping means that each character is replaced by another character defined in the translation file. The result depends on the translation file. Example: the text is remapped to make it readable by MSDOS machines if you load the 'AmigaToMSDOS' table. Without this conversion some characters would have been displayed incorrectly if the file were transferred to a PC. Remapping may be lossy. Example: the "@" character can not be mapped to a similar MS-DOS character (there is no such character in the PC character set). "c" is used instead.

## 1.87 search/function list

search/function list of: SEARCH MENU

QuickFunc

Presents a new window listing recognized "markers" found in the document. Click at one of those entries to move the cursor to where the "marker" has been found. Recognized markers depend on the document's filetype (the file name suffix): C-Functions are listed if the document is a C source code. Other recognized filetypes include Pascal (listing procedures/functions), AutoDocs, Assembler, etc. GoldED uses "marker parsers" to handle different filetypes. Use the "mode" gadget of the QuickFunc window to modify filetypes recognized by the editor (file name patterns) or to add new external marker scanners.

---

Marker detection quality depends on the scanner. For example, the C scanner shipped with GoldED recognizes only a very common type of source code formatting (function name and arguments in one line, function type in the previous line). You may add additional scanners to adjust the system to your likings (search/Function find).

Marker parsers are selected by GoldED automatically depending on the file type of the text (derived from the file name suffix) and the marker configuration (marker pattern). Marker setup affects the reference system, too (extras/options local/reference), since marker scanners are used to create the reference database.

## 1.88 search/function find

search/function find

Move the cursor over an expression (e.g. a function name) and execute this function: the cursor is moved to the "expression definition" if one available in the same document. Internally, an invisible list of all markers found in the text is created (see QuickFunc) and searched for the expression. Marker list creation depends on the text type and the QuickFunc setup.

Technical information: programming marker scanners

GoldED supports external marker scanners: select the "mode" gadget (QuickFunc window) to open a setup requester, then load a scanner using "add". The scanner is LoadSeg()'ed by GoldED, i.e. it is expected to be an executable. External scanners are called for every line of a document. It will receive the address of a line's text (char \*\*) in A0. The line's length is available in D0, the line number is available in D1. The scanner will have to examine this line. It is expected to return NULL if the line is of no interest from the scanner's point of view (e.g. a scan handler looking for #defines will return NULL if the line doesn't contain a #define). Or it may return the length of a result string to be displayed within the jump table. Set the string pointer (address available in A0) to the address of a NULL-terminated result string in this case. Example source code is available in the golded:developer/examples/scanner drawer.

## 1.89 VIEW MENU

VIEW MENU  
menu tree of view menu

view/new window	view/syntax highlighting
view/go to line	view/debugger
view/go to top-bottom	view/toolbar
view/go to modification	view/window
view/position store	view/window next
view/position recall	view/window previous

view/fold  
view/preview

view/hidden buffers

## 1.90 view/new window

view/new window of: VIEW MENU

Opens a further text window. Window size and window position are determined by the local configuration ( extras/options local/display ): the window is centered on screen, aligned or enlarged. You may specify a fixed position/size, too: disable AutoArrange and disable AutoCenter. Resize the window according to your wishes, then save the display configuration.

## 1.91 view/go to line

view/go to line of: VIEW MENU

Moves the cursor to the line you specify (1, ...). Check the 'unfold' gadget to enable absolute line numbers (folded lines are counted). Folded lines are ignored if this gadget is unchecked.

## 1.92 view/go to top-bottom

view/go to top-bottom of: VIEW MENU

Moves the cursor to the document's first line or to the document's last line. The cursor is moved to the first text line if closer to the text end so far (and vice versa).

## 1.93 view/go to modification

view/go to modification of: VIEW MENU

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

## 1.94 view/position store

view/position store of: VIEW MENU

Records the position of the cursor & the view area of the text in the window. Use view/position recall to restore the cursor position and view area. You

---

may store five positions for each text window separately.

## 1.95 view/position recall

view/position recall of: VIEW MENU

Recall cursor position stored by view/position store .

## 1.96 view/fold

view/fold of: VIEW MENU

Folding

INTRODUCTION

One of GoldED's most useful features is its folding capability: Folding means hiding some lines of a document temporarily. The lines are replaced by a single (highlighted) line called the "fold comment". Folding simplifies editing large files: you don't get lost in thousands of lines if you fold parts you are (currently) not interested in. Unfold the ones you are working on. Nest folding is supported: folds may contain further folds.

How to fold lines

Enclose the section to be folded (hidden) into 'fold markers'. Fold markers are plain character sequences. Default sequence is /// (which is recognized as a comment by most C compilers). Example:

```
/// "important function"

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

///
```

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

```
> important function
```

Press the HELP key again to unfold the fold. There number of folds used in a text is not limited. Using some folds will increase the performance of most functions (e.g. line insertion) though using a large number of folds may decrease the performance.

---

Fold markers may be set to any string of up to 10 letters. The shorter the faster. Nested folding is supported only if different markers for beginning/end of a fold section are used (see `extras/options local/misc`). You should choose fold markers carefully: we recommend to use a sequence regarded as comment by your compiler. Or embed the fold markers into comments.

Fold comments (displayed instead of the folded text) are write-protected. You can not edit those 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 edit it (or search text in it). Block operations ( `block menu` ) don't modify folded sections, however, folds are not ignored completely: 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.

#### FOLD ALL

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.

#### UNFOLD ALL

Scans text for folds. All folds (and sub-folds) are unfolded.

## 1.97 view/preview

view/preview of: `VIEW MENU`

Toggles display mode of current window from 'normal' mode (standard text font used) to 'preview' mode (preview font used; usually a small font to increase the overall view). See `Preview` for more details.

## 1.98 view/syntax highlighting

view/syntax highlighting of: `VIEW MENU`

Toggles syntax highlighting on/off: GoldED color-codes syntax elements recognized by syntax parsers (e.g. reserved words of a programming language) if syntax highlighting is enabled (`extras/options local/syntax`).

## 1.99 view/debugger

---

view/debugger of: VIEW MENU

Debugger

Toggles the breakpoint display of the current window on/off. Requires a source level debugger recognizing GoldED. Usually debuggers recognizing GoldED enable/disable breakpoint displays on their own. This menu allows you to override debugger control. Breakpoints usually are created by the compiler or debugger and set by the user to either active or inactive. Click at a breakpoint to toggle its state. Breakpoints can be deleted (SHIFT-click).

A typical debugger session using GoldED as debugger frontend: Compiler and debugger are started by the compiler environment. The debugger assumes control over GoldED if it has been prepared for GoldED. Breakpoint displays automatically appear in all text windows connected to your project. You'll use the debugger window to step through the program or inspect variables and the breakpoints displayed by GoldED to tell the debugger where to stop execution. Breakpoints are hidden at the end of the debugger session automatically. GoldED allows you to modify buffers during a debugger session. You can correct errors immediately. Unmodified buffers are simulated towards the debugger: the debugger is not confused, breakpoints remain usable.

Information about breakpoints optionally is stored in files so that you can restore the breakpoints quickly to a specific state during the next debugger session ( extras/options local/misc ): The editor optionally saves breakpoint information to files named like the original text with the suffix set to ".bpt". These files are created automatically if you save a text containing breakpoints and loaded automatically if found while loading a text. They are deleted automatically if you save the text again without defined breakpoints (you may prefer to manually delete). Please note that it depends on your debugger whether pre-set breakpoints loaded from a bpt-file are accepted during a debugger session: the debugger needs to support coordination/synchronization of pre-set breakpoints with the list of available breakpoints produced by the compiler.

## 1.100 view/toolbar

view/toolbar

Toolbar

Shows/hides the toolbar. Look and feel of the toolbar and the functions assigned to the buttons are configurable: extras/options local/gadgets .

## 1.101 view/window

view/window

This menu offers window-oriented functions:

---

```
view/arrange vertical
view/arrange horizontal
view/window maximize
view/window center
view/window zip
view/remember size
```

## 1.102 view/window arrange

view/window arrange of: VIEW MENU

```
view/arrange vertical
view/arrange horizontal
```

Arranges windows on screen while avoiding overlapping windows. The windows are made to share the display without overlapping. The screen's title bar is kept uncovered (correct recognition of the title bar is not possible if the screen has not been dragged down). You may assign extra weight to the current window using the "weight" gadget of extras/options local/display . This function recognizes AutoScroll screens (screens larger than the display): the window(s) are arranged within the visible section only. However, you may instruct the editor to use the full screen (display configuration, full screen option). Screen areas to be ignored while arranging windows may be specified ( extras/options local/display , margins). Adjusting those margins allows you to keep important display elements visible.

## 1.103 view/window maximize

view/window maximize of: VIEW MENU

Enlarges the current window to cover the visible display area. User defined margins and the position of the screen's title bar are considered (see view/window arrange ).

## 1.104 view/window center

view/window center of: VIEW MENU

Centers the current window. User defined margins and the position of the screen's title bar are considered (see view/window arrange ).

## 1.105 view/window zip

view/window zip of: VIEW MENU

Resizes 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 alternative window size is preset by GoldED to the minimum window size the editor can handle.

### **1.106 view/remember size**

view/remember size of: VIEW MENU

Size and position of the current window are used as default window dimensions for windows sharing the display configuration of the current text. You can save default dimensions permanently using extras/save preferences . Please note that other configurable options of GoldED may override default window dimensions (e.g. AUTOARRANGE ).

### **1.107 view/window next**

view/window next of: VIEW MENU

Activates the next window (based on window creation order). A very handy function: you don't need to grab the mouse for window activation. Simply flip from one window to another window using this command.

### **1.108 view/window previous**

view/window previous of: VIEW MENU

Activates the previous window (based on window creation order). This menu is the counterpart for view/window next .

### **1.109 view/hidden buffers**

view/hidden buffers of: VIEW MENU

HIDE

Hides a text buffer, i.e. closes the window but keeps the document. Useful to have several text files present without overloading the display. Use BUFFER SHOW to reopen the window. Freezing the last window leaves you without menus and without a GoldED screen. You'll then have to use the hotkey to reopen the display: right ALT plus right SHIFT plus RETURN. Or

---

the commodities exchange program of your Workbench.

#### BUFFER SHOW...

Asks you for a hidden buffer to be reopened. This functions tries to restore the old window position/size.

#### BUFFER SWAP...

A list of hidden buffers is display. Select a buffer to be reopened. The current window is hidden instead.

#### BUFFER SWAP (NEXT)

Reopens the next hidden buffer. The current window is hidden instead.

#### BUFFER SWAP (PREV)

Reopens the previous hidden buffer. The current window is hidden instead.

## 1.110 EXTRAS MENU

EXTRAS MENU  
menu tree of extras menu

extras/project filter	extras/redo
extras/search file	extras/AmigaDOS shell
extras/search file...	extras/HiSpeed
extras/line swap	extras/files
extras/line double	extras/insert text
extras/line remove	extras/options local
extras/line insert	extras/options global
extras/undo	extras/save preferences

### 1.111 extras/project filter

extras/project filter of: EXTRAS MENU

Global search

Displays a requester to enter the text to search for. Searches all files added to the project list ( extras/options local/project ). File names are highlighted if the pattern is found in the file (OS3.0). The file name is removed from the list if the pattern is not found in the file (OS2.0). You may choose between fast case sensitive and slow case-insensitive searching using the appropriate checkmark gadget. Doubleclick at a filename to load the corresponding file. Hold the SHIFT button down to toggle selections (OS3 only). The list displayed by this requester is a copy of the project list, adding files or removal of files doesn't affect the project list.

List access (programmers only)

External programs may read the project list by sending the QUERY command to GoldED's ARexx port (QUERY PRJLIST). The editor returns a pointer to a standard exec list. This pointer stays valid until the next ARexx command is sent to GoldED ( LOCK / UNLOCK protocol required). Example source code is available in the developer drawer ("golded:developer/examples/project"). The list may not be modified directly (use the PROJECT command to edit the list).

### 1.112 extras/search file

extras/search file of: EXTRAS MENU

File hunter

Extracts a file name from the text under the cursor. The editor understands several different file name delimiters (e.g. <...> or quotations marks). It defaults to extracting the full word under the cursor if no known delimiters are detected. The editor will then try to find the file on your disk(s). All locations specified by the hunter setup ( extras/options local/hunter ) are scanned.

### 1.113 extras/search file...

extras/search file... of: EXTRAS MENU

Function to locate a file on your disk(s) ( file hunter ); enter the file you are looking for.

### 1.114 extras/line swap

extras/line swap of: EXTRAS MENU

Swaps current line with next line.

### 1.115 extras/line double

extras/line double of: EXTRAS MENU

Doubles current line (faster than using block/copy ).

---

## 1.116 extras/line remove

extras/line remove of: EXTRAS MENU

Deletes current line. The line is not lost - it is put to a pick-push ring buffer (last in, first out). You may reinsert it anywhere else using extras/line insert . This function is assigned to the keyboard, too (CTRL-DEL). Note: keyboard access (CTRL-DEL) will give you a much better response time than menus due to the Amiga's OS (keyboard queue settings unfortunately don't affect menu shortcuts). Example usage: Delete 3 lines, move to another line and call extras/line insert three times to reinsert the lines. The pick/push buffer holds up to 50 lines.

## 1.117 extras/line insert

extras/line insert of: EXTRAS MENU

Reinserts the last line of the pick/push buffer before current line. See extras/line remove for further explanations.

## 1.118 extras/undo

extras/undo of: EXTRAS MENU

undo & redo

Undos last operation if undo has been enabled ( extras/options global/misc ). You can undo the undo using extras/redo immediately. You won't be able to redo undos if you modify the text. The number of undoable operations depends on how much memory you have assigned to the undo mechanism (see extras/options global/misc ). All operations including block-related functions, formatting and loading can be undone. However, depending on the selected undo mode ( extras/options global/misc ), some operations can not be undone separately. In standard mode, all modifications within a single line count as one undo step. In "High" mode, all major changes (e.g. "delete until end of line") within a line can be undone separately while minor changes (single keystrokes) still are collected as one undo step. In "Auto" mode, all modifications can be undone separately. A garbage collection has been added to reduce memory consumption in "Auto" mode: multiple "old" minor steps are converted to one major step automatically (occasionally). Memory consumption created by the undo system can be checked using project/information (undo bytes display).

## 1.119 extras/redo

extras/redo of: EXTRAS MENU

Undos the last undo. This command can only be used immediately after an undo: All redo information is discarded once you start editing the text.

## 1.120 extras/AmigaDOS shell

extras/AmigaDOS shell of: EXTRAS MENU

Opens a shell window on GoldED's screen. The window is arranged on the screen: it will open within the visible display area of overscan screens.

## 1.121 extras/HiSpeed

extras/HiSpeed of: EXTRAS MENU

Runs the external HiSpeed printer utility. The document name is passed as argument to this program. Note: HiSpeed will read the file, it will not read the buffer maintained by GoldED. Save the file first if it has been modified.

## 1.122 extras/files

extras/files of: EXTRAS MENU

DELETE FILE

Asks you for a list of files to be deleted. Hold down the SHIFT button to select multiple files. You are asked for confirmation before files are deleted. It is not possible to delete delete-protected files.

RENAME FILE

Allows you to rename a file. Select a file and enter the new name.

CREATE DIRECTORY

Creates new drawers. Enter the drawer name. Attempts to re-create existing directories result in an 'object in use' error.

## 1.123 extras/insert text

---

extras/insert text of: EXTRAS MENU

INSERT DATE

Inserts the current date (e.g. "Boston, 3/5/96") at the 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 saves variables to the 'env:' directory only. Unfortunately this directory usually is located in the RAM disk, ie. your modifications are lost after a reset. Copy env:usertown to envarc:usertown to avoid losing modifications.

INSERT TIME

Inserts current time into the document.

INSERT PATH

Allows you to insert a file name at cursor position (a file requester supports fast selection).

## 1.124 extras/options local

extras/options local

OPTIONS LOCAL

extras/options local/project	extras/options local/misc
extras/options local/hunter	extras/options local/API
extras/options local/reference	extras/options local/tabs
extras/options local/printer	extras/options local/layout
extras/options local/words	extras/options local/GUI
extras/options local/templates	extras/options local/display
extras/options local/menu	extras/options local/syntax
extras/options local/mouse	extras/options local/gadgets
extras/options local/keyboard	extras/options local/indent

This multi-page notebook requester provides access to the local configuration options ("environment"): configure the menu, the keyboard or the display. Environments are assigned to documents automatically: the editor selects environments based on the document type (derived from the file name). For example, you could instruct GoldED to use an environment designed for editing C sources if a C source code is loaded. Creation of environments is handled by extras/options global/filetypes (as is selection of environments depending on the filetype). GoldED is shipped with a single preconfigured environment called "TEXT". Further preconfigured environments are available at the GoldED WWW support site ( WWW support ).

It is important to understand that all environments consist of smaller units called "presets". Some presets have been mentioned above already: menu configuration, keyboard configuration or display configuration to name a few. The local configuration requester lists all presets assigned to the current document (forming its environment). GoldED allows you to share presets between environments and thus supports shared resources and centralized

maintenance of configuration. That's why the term "local configuration" is slightly misleading: It's your decision to use presets locally only or to share them among multiple environments ( extras/options global/filetypes ).

Technical information: presets are stored in "golded:registry/presets/". Environments are saved to "golded:registry/registry". These files and directories may not be modified manually (don't copy files to these directories, don't delete or rename files). Ignoring this warning could destroy the registry and all configuration information. Installation scripts should use the tool "golded:tools/regedit/regedit" to install new presets and new environments.

### 1.125 extras/options local/project

extras/options local/project of: options local

Project management: Add all source files of your current project to this list. The project list should be viewed as a basis for external project management utilities. No project management capabilities have been added to GoldED. GoldED usually will not access this list with exception of the grep command (extras/project filter).

### 1.126 extras/options local/hunter

extras/options local/hunter of: options local

Used to tell GoldED where to search for files if the file location functions are used ( extras/search file ). Subdirectories are examined, too, if the "recursive" gadget is selected.

### 1.127 extras/options local/reference

extras/options local/reference of: options local

Configuration requester for the reference system. The reference system is based on databases: databases are collections of keywords accessed by the editor if the user asks GoldED to display a help file ( search/reference ). For each known keyword the path to a help file is stored in the databases. Databases are created by GoldED ("create" gadget). All you have to do is to provide a list of help files: source codes, AutoDocs, HTML files (or whatever you consider useful). You may add directories to the list, too (subdirectories are scanned). GoldED will scan these files and extract keywords to build databases ("create" gadget). Database creation may take a long time depending on the number of files to be scanned. Keyword extraction is performed by parsers made for parsing special filetypes (e.g. C sources are parsed by a C parser extracting function definitions). External parsers may be added in order to increase the editor's capability to understand more

---

file types. Parsers are added and configured using search/function list . GoldED is shipped with parsers for C source codes, C header files, C include files, assembler source codes, Pascal source codes, BASIC source codes, AmigaGuide files, AutoDocs and HTML documents.

## 1.128 extras/options local/printer

extras/options local/printer of: options local

Printer

Provides additional configuration options (basic configuration is provided by the Workbench printer preferences utility). Supports all Workbench printer drivers as opposed to extras/HiSpeed supporting PCL printers only.

INIT

The contents of the init string gadget are send to the printer after basic printer initialization. The init string may be used to pass additional control codes to the printer device. The init string is interpreted (see magic codes ), non-ASCII codes can be added. Usually the printer device doesn't allow you to pass printer-specific codes to the printer - only printer device control codes are available (translated to printer specific control codes by the printer device). However, a special printer device command called "aRaw" supports sending embedded binary data to the printer. Syntax: 27 [ <number of bytes> 34 r. The following example init string sends the string 0123456 (seven letters) to the printer:

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

SPOOLER

Check the spooler gadget to have data sent to the printer in the background: Printing won't block text input, you can continue to edit documents while printing. A copy of the original text (spooler usage increases memory consumption) is sent to the printer, modifications added while printing will not affect the output.

## 1.129 extras/options local/words

extras/options local/words of: options local

Dictionary configuration. Several input aids provided by GoldED depend on the dictionary composition: AutoCase (automatic case correction) and expression completion ( APC ).

AutoCase

Text input is spell-checked (correct case ?) if this option has been enabled. Checks are performed after a line has been modified. Misspelled expressions are replaced by the correct form found in the dictionary.

## General hints

Don't let the dictionary grow too big. A large dictionary reduces usability of the APC function (you'll have to type more letters to ensure unique identification). Add unique expressions only. Add spaces to the dictionary if the expression usually is followed by a space character; example: add "int " (C type: integer) and not "int" (prevents GoldED from converting INTERNATIONAL to international). Use this strategy for brackets, too: Add "Open(" to the dictionary, not "Open".

### 1.130 extras/options local/templates

extras/options local/templates of: options local

Templates configuration ( Templates ). Templates are search patterns expected in the input stream. If a known template (a keystroke sequence) is detected in the user input the "action" assigned to the template is performed (provided that templates detection has been enabled: layout/templates ). Only single words may be added as templates; no spaces. Action configuration is similar to event configuration of menu items or keyboard bindings (see event definition ). Doubleclick at one of the templates displayed in the list to modify event definitions.

### 1.131 extras/options local/menu

extras/options local/menu of: options local

#### Menu configuration

Menu titles (left list), menu items (middle list) and subitems (right list) are displayed and can be modified. The number of menus, items and subitems is limited by the Amiga operating system: for example, you may not add more than 32 items to each menu title. Doubleclick at any item to enter item configuration ( event definition ). Item configuration is similar to keyboard configuration, mouse configuration, etc. The guide gadget allows you to specify a guide file used while processing MenuHelp requests.

#### File list

All documents currently loaded (hidden buffers included) are listed in the menu if this option is used, providing a fast document selection method. Holding the SHIFT key down while selecting a hidden buffer in the file menu will show the hidden buffer and close the current window. Activation of the file list feature affects performance: The OS functions used to maintain the list are slow (updates of the file list are required if a window is closed or a text is loaded).

---

## 1.132 Event definition

### Event definition

The event definition requester is used to configure menu items, the keyboard, mouse keys and several other resources generating events if used. It allows you to assign internal commands, macros, text, etc. to events (e.g. a menu selection: extras/options local/menu ). Add the command, script or string (use quotation marks) to the action list and set the action type to command, script, string, etc.

DOS, AREXX and TEXT action entries are interpreted: predefined keywords are replaced by a corresponding "value" at execution time (quote a string if you don't want it to be interpreted). The following keywords are recognized:

Magic codes

```
"text" ..... raw text
\n ..... inserts line feed
\t ..... inserts tab
\" ..... inserts quotation mark
\DATE ..... \ inserts value of internal variable (e.g. DATE)
$HOME ..... $ inserts value of environment variable (e.g. HOME)
%27 ..... % inserts ASCII code (e.g. ASCII code 27)
```

The number of actions that can be added to event definitions is not limited. Mixed action types (e.g. internal command to save a text followed by a macro call) are supported. ARexx macros are started asynchronously: event processing will not stop. Event configuration should not depend on assumptions about if or when a macro has been processed by the ARexx server.

PATH

Sets current path for event processing. Magic codes are supported.

OUTPUT

Sets output path/device. Magic codes can be used in this gadget. Example: con:0/0/640/400. The default console is used if this field is left empty ( extras/options local/GUI ). PATH and OUPUT settings affect DOS and AREXX actions only.

SHANGHAI

Windows meant to appear on the default public screen (Workbench screen) are redirected to GoldED's screen if this option is set to a positive value. Units are seconds. Redirection is active for the selected interval only. Not all windows can be redirected.

ASYNC

Asynchronous and synchronous DOS actions are supported. Synchronous DOS actions block event processing until the DOS command terminates.

HOTKEY

Menu shortcut definition (ignored if the event is not assigned to a menu item). Many shortcuts are consumed by the standard menu already. Click at the arrow gadget to make GoldED list unused shortcuts. Only ASCII characters (ASCII codes below 128) are listed. National characters are not listed. Uppercase shortcuts and lowercase shortcuts are listed separately. The text you enter into this field will not work as a menu shortcut if you enter more than one character: Instead, the text appears in the menu literally (OS3 required). Recommended usage is to notify the user of keyboard bindings that could be used as a menu shortcut.

#### VARIABLE

Variable selection (ignored if the event is not assigned to a menu item): A checkmark is added to the menu. The (boolean) variable value is used to control the checkmark state. This is just a rendering operation, usage of the menu will not change the checkmark state until you add internal commands to the event setting the variable (e.g. the SET command).

#### GUIDE

You may assign pages of AmigaGuide documents to events in order to provide online help. For example, online help is displayed if the user holds down the HELP key while selecting a menu ([menu help]). Enter the name of an AmigaGuide page into the guide gadget. The page name usually refers to the menu's AmigaGuide documentation ( extras/options local/menu ). However, you may specify a specific AmigaGuide file, too: enter the name of the guide file followed by "@" followed by a page name.

You may assign a short help text directly to an event, too: the contents of the guide gadget are interpreted as text (and not as a reference to an AmigaGuide document) if put in quotation marks. The help text of toolbar events is displayed as "bubble help" if the user moves the mouse pointer over a toolbar symbol for a few seconds.

### 1.133 extras/options local/mouse

extras/options local/mouse of: options local

Mouse configuration. This requester is used to configure the mouse buttons. The left mouse button and the middle mouse button (if available) can be reconfigured. Event definition is similar to menu/keyboard configuration ( event definition ).

### 1.134 extras/options local/keyboard

extras/options local/keyboard of: options local

Keyboard configuration. The keyboard requester is used to map actions to keys or to key-qualifier combinations. GoldED recognizes the qualifiers ALT, CTRL and SHIFT or a combination of these keys. The system's default keymap is used

---

for unconfigured keys: The "A"-key inserts "A" at the current cursor position if unconfigured. Control keys are initially undefined, ie. unconfigured cursor keys will not move the cursor, etc. GoldED is shipped with all control keys configured to useful defaults. Configuration of a key(-combination) is easy: Use the "record" gadget to add a key to the list of configured keys. Doubleclick at any of the listed key definitions to modify it. Keyboard event definition is similar to menu configuration: event definition . Some key combinations are consumed by the OS and are not available for remapping. These keys are called 'dead keys'. For example, ALT-G doesn't insert a character but determines how to modify the next character: ALT-G + "a" appears as "à". The following keys are dead keys:

```

`          ALT-H
ALT-F     ALT-J
ALT-G     ALT-K

```

### 1.135 extras/options local/misc

extras/options local/misc of: options local

General configuration options.

Fold start, fold end

Fold marker strings (see folding ). Different markers are required if nested folding (folds within folds) is used. Otherwise the same marker should be used for fold start and fold end.

AutoFold

Folds found in a text are automatically folded while loading if this option is enabled (before the text is displayed for the first time).

XPB compression mode

The XPB listview (arrow gadget) presents a list of available XPB compressors, found within the libs:compressors subdirectory. The compressor you select is used if saving XPB files ( project/save as XPB ). Additionally you may set the efficiency for compression (0% to 100%). Please read the original XPB documentation. Some XPB compressors require a password set using extras/options global/misc .

Create \*.bak

The editor backups old copies of a file before saving a new version over it if 'create \*.bak' is enabled (otherwise the old version is overwritten). Backups are saved to the backup path you name (backup path gadget) or to the document's directory if you don't specify a backup path. The suffix ".bak" is appended to the file name.

Create \*.info, \*.info file

This option decides if icons are created for the files saved by GoldED. (existing original icons are not overwritten).

### Create \*.bpt

Information about defined breakpoints is saved permanently if you enable this option (so that breakpoints can be restored quickly to a specific state during your next debugger session). Breakpoint information is saved under the name of the original text with the suffix set to ".bpt". These files are created automatically if you save a text containing breakpoints ( Debugger ) and they are loaded automatically if the text is loaded ( project/open ).

### Default tool

Default tool inserted into icons created by GoldED.

### File protection bits (Readable, Writable, Executable, Deletable)

These gadgets are used to set the default bits for a new text (created by view/new window ). Use project/protection to modify protection bits of open buffers.

### Overwrite

Existing copies of a document are overwritten without further notification (confirmation requester) if this gadget is checked.

### Save Tabs

Leading spaces of each line are replaced by tabs while saving if this option is enabled. Saving tabs instead of spaces reduces the file size. The number of spaces saved as one tab code can be adjusted; default is eight characters ( extras/options global/misc ). This option should not be used unless you are very short of disk space (GoldED can load files without tabs faster).

## 1.136 extras/options local/API

extras/options local/API of: options local

API interface configuration: Select the clients you would like to use locally. Clients are editor extensions providing additional features. They are added to GoldED using a dedicated library-based protocol. Add the API clients to the list and - if required (read the documentation of the API clients you add) - enter a command string in the "arguments" gadget. API example source codes are available in the "golded:developer/examples/api" drawer. Specifications of the API interface can be found in "golded:developer/api". API clients provide ways and means to add complex new functions to GoldED. A few simple examples are shipped with GoldED:

### MIRROR.API

The mirror client highlights the matching bracket while typing: move the cursor over a bracket to have the matching bracket highlighted. Highlighting is not available while the block marker is used. Direct

---

neighbours are not highlighted to avoid confusion with the cursor.

#### REXX.API

Command set extension

The API interface supports command set extension. The REXX.API example adds the commands DISPLAYBEEP and ABOUT to GoldED. These commands can be used in menus, keyboard configuration, etc.

#### STARTUP.API

This API client runs a macro if activated; it can be used to automatically start external programs. Enter the macro to be executed in the arguments field.

#### QUICKINFO.API (idea: Jochen Becher)

This client displays information for the word under the cursor in the status bar. Example usage: have function prototypes displayed. Enter the dictionary file with the explanations in the arguments field (e.g. golded:tools/quickinfo/standard.words). Each line of that file defines a keyword and - separated by a space - keyword-related information to be displayed if the cursor moves over that word. Keywords may contain the letters "a-Z", numbers, the underscore and a trailing paranthesis ". The dictionary file has to be kept sorted (case-sensitive). Use GoldED's BLOCK SORT CHKCASE command to resort a dictionary file after modifications.

Technical information: API clients are libraries. The AmigaOS doesn't allow renaming libraries: Renamed API clients can not be used. Be careful to add the correct name to the list of API clients (case sensitive).

## 1.137 extras/options local/tabs

extras/options local/tabs of: options local

Tab configuration requester. GoldED supports several tab modes: regular tabs are spaced at a constant distance (tab distance gadget). Standard tab distance of Amiga editors is 4 spaces. Fixed tabs can be set to arbitrary positions: add the desired tab stops to the list. Fixed tabs should be used for tables or assembler source codes. Last but not least dynamic tabs are available:

#### Dynamic tabs

In dynamic mode tab stop positions depend on the context: GoldED will examine the last line(s) to determine appropriate positions. Regular tabs are used if the context doesn't indicate possible positions.

All tab modes mentioned above can be used in light mode (cursor is moved to the next tab stop) or in editing mode: spaces are inserted while moving the cursor to the next tab stop, indenting the text after the cursor. GoldED doesn't use tabs codes (ASCII 9) internally nor will it assign a special meaning ("indention") to this code. Other Amiga editors may use tab codes to indent text. GoldED expands those tab codes to spaces while loading (see

---

project/open ). Usually one tab code is expanded to eight spaces. You may modify this value ( extras/options global/misc ). A few programs use tab codes in their configuration files. Those tab codes usually may not be expanded to space. Solution: use the RAW option of the OPEN command to override tab expansion.

### 1.138 extras/options local/layout

extras/options local/layout of: options local

Layout configuration. These settings affect several functions of GoldED ( Layout menu ). Layout borders (left margin, right margin) set by this requester are related to formatting functions only, they usually don't restrict cursor movements. Two WordWrap levels are available: in standard mode linefeeds are inserted automatically if the user input exceeds the right margin. Improved mode ("reformat" enabled) additionally reformats the text until the end of the current paragraph. Enable "reformat hyphens" if you want to have hyphens at the end of lines removed if appropriate. This feature avoids hyphens within lines after reformatting paragraphs: word parts are concatenated without a hyphen.

### 1.139 extras/options local/GUI

extras/options local/GUI of: options local

Configuration of user interface related options

#### FAST SCROLLING

GoldED will speed up scrolling after the cursor has reached the borders of a text window if the FastScroll gadget is checked. You might want to turn this feature off in monochrome mode (provided you own a fast A4000) to slow down the display.

#### ERROR BEEP

Error beep volume.

#### SCROLL BORDERS

Sets the minimum cursor-to-window-border distance triggering scrolling to make more text visible.

#### WHITE SPACE

Tells GoldED what characters should be handled as if they were spaces (white space characters). This definition is used by several functions. For example, including "(" to the list would make the 'jump to next word' function (SHIFT CURSOR RIGHT) consider the "(" as word separator. Format of this string: The list of white space characters may consist of ASCII codes,

---

code ranges or strings, separated by colons (no spaces). Example:

```
0-" ",128-160,".,;()".
```

#### EOL WRAP

Enable the EOL-wrap mode to make the cursor jump to the beginning of the next line if the cursor passes the last character of a line (during user input).

#### CONSOLE

Standard output device: used while processing menu/mouse/keyboard events (if no other output device has been specified for the event). The console description string is interpreted, you may use the magic codes within the console description. Defaults to a reasonably-sized window on the editor's screen.

#### EXCLUDE CURSOR

Determines whether the cursor is part of the block while marking. You might want to adjust cursor color settings ( extras/options local/display ) after changing this option in order to visually separate block and cursor.

#### PARENTHESIS CHECK

Enable ()-check if you want the current line checked for correct use of brackets if accepted (cursor leaving line); see search/check .

## 1.140 extras/options local/display

extras/options local/display of: options local

Display mode

Font

Display configuration: adjust screen resolution, fonts, window look and more. You shouldn't modify the display configuration if running low on (CHIP-)RAM: The editor might not be able to allocate enough RAM to reopen the screen. You may select different fonts to be used for the text, for requesters and for menus (if the editor is running on a custom screen). Fonts have to be fixed width fonts. Only the screen font may be a proportional font. Line spacing of the text font is variable; default is 0 (no extra spacing between lines). Fonts smaller than eight pixels (width or height; e.g. XEN, thinpaz) in text windows may cause problems on a few Amigas: random pixels appear in the text. This problem can not be fixed.

#### PREVIEW

You may specify a preview font: This font is used by GoldED if you switch a window to preview mode ( view/preview ). The preview font should be considerably smaller than the standard text font. Intended usage is to increase the overall view on the fly, thus avoiding to loose track while working on complex sections of a source code.

---

## ICONS

Enable/disable use of icons within requesters. Since the editor's icons have been designed with the OS2/OS3 color scheme they might not look that good if you use your own color scheme. Use this gadget to turn icons on/off.

## PENS

You may set the pens used by the OS (and GoldED) to render the user interface. Some pens are not adjustable under OS2.1 or earlier. For example setting the menu background pen used to render menus requires at least OS3.0.

## CHUNKY PIXEL

Many external graphics boards don't use a bitplane representation of graphics (as the current Amiga chipsets do) but a chunky pixel organization. Use the 'chunky mode' gadget to configure GoldED according to your hardware. Usually (chunky pixel gadget not checked) GoldED will try to speed up the display by restricting output to single planes - this will give you a considerable speed increase as long as a native plane-based Amiga chipset (e.g. the ECS chipset) is used. But it won't give you a speed increase at all if your external graphics board is based on a chunky pixel organization. In fact write-protecting planes might even slow down output, so switching GoldED to chunky mode might be a good idea if you own such a board. Be careful to have this gadget set properly to avoid loss of performance. Uncheck it if in doubt.

## FULL SCREEN

Functions related to arranging windows (e.g. view/window arrange ) usually determine the screen's visible display rectangle and try to arrange the windows within this area. Enable the full-screen option if you want to have the screen's real size considered instead. Useful if the driver software of your graphics board doesn't support the OS functions QueryOverscan() and VideoControl() required to read the overscan size.

## CENTER WINDOWS

Activate this option to have new windows opened centered on the screen (minus margins; see below).

## AUTOARRANGE

Windows are rearranged automatically after a window has been closed or while a new window is prepared if AutoArrange has been enabled (see view/window arrange ).

## WEIGHT

Used to assign extra space to the current window for window arrangement ( view/window arrange ). Choose a weight of two to make the current window appear twice as big as the other windows.

## MARGINS

---

These gadgets allow you to specify borders to be left free during window arrangement ( view/window arrange ).

#### SCROLL BAR

The bottom scrollbar may be turned on or off (turning the scrollbar off enlarges the text area).

## 1.141 extras/options local/syntax

extras/options local/syntax of: options local

### Syntax Highlighting

Syntax highlighting preferences: GoldED can color-code text and thus increase usability. Colors to be used are set by this requester. Typical usage of syntax highlighting is to have reserved words of a programming language highlighted. Syntax parsing is performed by external parsers selected using this requester: the editor is responsible for refreshing the display. The syntax parsers is responsible for parsing the text and for creating a syntax description. The highlighting scheme depends on the parser. Parser for C(++) sources, Assembler and HTML are shipped with GoldED. Example source is included, too, in case you want to write your own syntax parser (golded:developer/examples/syntax).

### Preparsing option

Syntax highlighting is prepared either immediately after loading (preparing enabled) or "on-demand": on-demand parsing reduces loading time and short-time RAM usage by restricting parsing to the visible lines. Invisible lines are not parsed until they are displayed for the first time. On-demand parsing will slow down the display refresh until all lines have been displayed at least once. Preparing should be disabled for complex (slow) parsers only.

### Parser names

Please consider that syntax parsers are standard libraries and that the AmigaOS doesn't support renaming of libraries: a syntax parser won't work after it has been renamed (parser names are case sensitive).

### Hardware

Usage of color highlighting should be restricted to a few colors only with respect to speed. Performance depends on the graphics hardware: a fast external graphics board is required to ensure sufficient performance. Amiga chipsets are based on bitplanes and therefor too slow. CPU speed is less important; suggested system is a 68030 CPU or better.

---

## 1.142 extras/options local/gadgets

extras/options local/gadgets of: options local

Configurable gadgets

GoldED supports two types of user-defined gadgets: text-only gadgets in window borders and toolbar gadgets. Add the button label respectively the path to a graphic (IFF) to the gadget lists. Doubleclick at a gadget definition to modify the action attached to the gadget ( event definition ).

## 1.143 extras/options local/indent

extras/options local/indent of: options local

Indentation configuration. These settings control the new cursor position after a linefeed (see return key ).

AutoIndentation

Enable AutoIndentation to re-use the current indentation.

SmartIndentation

SmartIndentation is context sensitive: the indentation scheme and the new cursor position depend on keywords and on the indentation scheme assigned to those keywords. For example, you can instruct the editor indent after the "IF" statement. The first and the last word of the cursor line determine the indentation scheme of the cursor line and the new cursor position in the next line. Use the listview gadget to edit or add keywords. Supported smart indentation schemes are:

```
-->    next line: cursor indentation
<--    next line: cursor outdentation
>>>   shift current line right
<<<   shift current line left
<<< -> shift current line left; next line: indentation
```

## 1.144 extras/options global

extras/options global

OPTIONS GLOBAL

```
extras/options global/filetypes
extras/options global/misc
extras/options global/variables
extras/options global/gui
```

Global configuration: Modify global options affecting all documents

---

independent of filetypes and local environments.

## 1.145 extras/options global/filetypes

extras/options global/filetypes of: options global

Basis of GoldED's configuration system are presets: several presets are assigned to a document (while loading) and control the menu, the keyboard configuration, the window look, etc. The presets control the environment of a text. Presets are selected from a large pool of available preset. Selection usually depends on the file name and thus indirectly on the file type: GoldED selects a set of presets based on the document's type. Combining presets to sets and assigning sets to filetypes is handled by this requester. Preset configuration is handled by options local .

### Filetypes

Add the filetypes to be recognized to the left list: click at any of the existing entries (preferably the last entry in the list - that's the default filetype) and adjust the trigger pattern displayed below the list. The trigger pattern is a standard DOS file pattern. It determines when the environment is activated. Example: setting the pattern to "\*.c" tells GoldED to use the new environment if editing C source codes. Please consider that the editor internally always uses expanded absolute file names (e.g. "dh0:s/startup-sequence", not "s:startup-sequence"). Filetypes are priority sorted: The list is scanned from top to bottom while determining the type of a text. The list's last entry is used if no match is found. Files types don't necessarily have to be DOS patterns: add any name you like. However, automatic file type detection is available for pattern-style filetypes only. Other filetypes require dedicated filetype selection: the Quickstarter provides a FILETYPE option to select a filetype. GoldED supports file type selection via the SET command (e.g. SET TYPE=".c").

The preset combination associated with the new filetype is controlled by the properties listview: it lists one preset for each configurable subsystem of GoldED (menu, keyboard, display, ...). The pool of available presets is displayed by the right list. Use the arrow gadget to add a preset to the filetype's properties. Duplicate a pool entry to create a completely new preset. Presets should be shared between multiple environments (filetypes) if possible to use resources efficiently and to share unique resources (e.g. screens).

## 1.146 extras/options global/misc

extras/options global/misc of: options global

Global configuration options.

RESIDENT

---

GoldED supports HotKey activation: The editor is not removed from memory even after the last window has been closed if hotkey activation is enabled (resident gadget checked). Instead, it will wait for a hotkey combination (right SHIFT & right ALT & RETURN) before it attempts to reopen its screen. HotKey activation will give you a quick response time though it will consume some memory, too. You might want to disable this feature on systems with less than 3 MB RAM. Use project/quit all & exit to remove the editor completely from RAM. The commodities exchange program of your workbench can be used to reactivate or unload GoldED, too.

#### AREXX DEBUGGER

Activates debug information output while processing incoming ARExx commands (commands arriving at the ARExx port are displayed). Informations are printed to the shell window GoldED has been started from. Simplifies macro development.

#### DISABLE FONT CACHE

Deactivates CPU cache while generating fonts. On some machines random pixels appear in text windows, especially if using narrow fonts (width below eight pixels, e.g. xen, thinpaz). This is probably a bug in the operating system. Enable this option and restart GoldED. Don't use fonts smaller than eight pixels ( extras/options local/display ) if experiencing display problems.

#### NUMPAD = MOVEMENT

Enables/disables special configuration of numpad keys: numbers (in standard mode) or cursor control commands.

#### UNDO RAM WARNINGS

The editor usually will notify you if the backup data generated by the undo system doesn't fit into the undo buffer: you will not be able to undo the last operation. Increase the undo buffer size (see below) if undo warnings appear frequently. No warnings are displayed if you uncheck this gadget.

#### REDUCED CHIP-RAM USAGE

Allows you to select a window refresh strategy for text windows. A window refresh is required if hidden parts of a window are made visible. You can not modify the refresh strategy for existing windows: this option affects new windows only. Two strategies are available: refresh performed by the operating system or refresh performed by GoldED.

Refresh performed by OS: REDUCED CHIP-RAM USAGE disabled

The operating system will refresh the windows using CHIP RAM to buffer hidden parts. Advantage: fast. Disadvantages: high CHIP RAM consumption. Performance suffers if the number of open windows increases.

Refresh performed by GoldED: REDUCED CHIP-RAM USAGE enabled

GoldED will refresh the windows. Advantages: reduced CHIP RAM consumption. Performance doesn't depend on the number of open windows. Disadvantage: Delayed refresh if GoldED is busy.

---

## ALLOW DUPLICATES

Enable ALLOW DUPLICATES to permit loading the same file multiple times. You'll be notified if a file has already been loaded. GoldED will immediately activate the existing copy if duplicates are not allowed.

## BACKUP [min]

## AutoBackup

Sets interval for automatic backups (0 deactivates backups). Backups are written to a configurable backup directory ( extras/options local/misc ). You may instruct the editor to display a confirmation requester before creating backups ( extras/options global/gui ).

## PASSWORD

Global password. Used while reading/saving encrypted XPK files (XPK compression mode is selected via extras/options local/misc ).

## UNDO MODE

Enables/disables the undo mechanism ( undo & redo ) and selects an undo mode: Standard mode, high mode and auto mode are available. Standard mode allows you to undo/redo all major changes. However, changes within lines can not be undone separately. In high/auto mode changes within lines can be undone separately ( extras/undo ).

## UNDO BYTES, UNDO STEPS

The editor backups text lines/parts you are about to change or delete if the undo mechanism has been turned on. Backup data is written to the undo buffer. The larger this buffer is, the more steps can be taken back using extras/undo . You can specify both, the undo buffer size and the maximum number of steps to be stored within the buffer. Old steps are deleted from the undo buffer if one of these limits is exceeded to allow storage of new information. The number of steps is a per-text limit while the undo buffer size is a global limit for the added undo memory consumption of all text buffers. Setting one or both of the limits to high values virtually disables the limit(s). Setting the size limit to low values has a bad effect on the editors performance. Avoid sizes below 100K. The undo buffer size is understood as "suggestion" only. Short time memory usage may exceed the limit. For example, buffer overflows created by extras/undo are tolerated to allow a redo extras/redo .

## TAB STANDARD

Informs GoldED about the standard tab distance used by other editors (in saved files). Amiga standard is that tabs indent 8 positions. A few editors assume that tabs indent four positions. You'll recognize documents saved by those editors immediately: excessive indentation of lines if loaded with GoldED.

---

## 1.147 extras/options global/variables

extras/options global/variables of: options global

Configuration of user-defined variables: you may add an unlimited number of user-defined variables to GoldED. These variables can be read ( QUERY ) and set ( SET ) as if they were built-in variables. They can be used to control the state of menu checkmarks ( extras/options local/menu ), too. User-defined variables have been created to support ARexx macro programmers: the main advantage compared to ARexx variables is that values are saved to the configuration ( extras/save preferences ).

## 1.148 extras/options global/gui

extras/options global/gui of: options global

### DATE FORMAT

Configures date strings (modifications don't affect existing date displays). Besides standard text, the following keywords can be used:

```
%a - abbreviated weekday name
%A - weekday name
%b - abbreviated month name
%B - month name
%d - day number with leading 0s
%D - same as "%m/%d/%y"
%e - day number with leading spaces
%j - julian date
%m - month number with leading 0s
%U - week number, taking Sunday as first day of week
%W - week number, taking Monday as first day of week
%w - weekday number
%x - same as "%m/%d/%y"
%y - year (two digits)
%Y - year (four digits)
```

### WINDOW TITLE

Format of window titles. The first "%s" found in this template is replaced by the file name at window creation time, the second "%s" is replaced by the filetype ( extras/options global/filetypes ).

### ASL FILE REQUESTER

Instructs GoldED to use the ASL file requester instead of the Reqtools file requester. Position and size of the ASL requester are saved to GoldED's configuration ( extras/save preferences ) while position and size of the Reqtools requester are set using the Reqtools preferences program.

### BRIEF MESSAGES

Enable "brief messages" to make GoldED write simple messages (no options to

---

be selected) to the window status bar instead of using a requester.

#### CONFIRM BACKUPS

Automatic backups have to be confirmed by you if this option is enabled ( extras/options global/misc ).

#### CONFIRM ERRORS

If set: Error requesters stop execution until the requester is confirmed.

#### BEEP

Error beep volume.

### 1.149 extras/save preferences

extras/save preferences of: EXTRAS MENU

Saves configuration ( options local , options global ).

### 1.150 MACRO MENU

MACRO MENU  
menu tree of macro menu

macros/session load	macros/sequence save
macros/session save	macros/sequence play
macros/macro edit	macros/sequence play...
macros/macro start	macros/macros
macros/macro execute	macros/execute command
macros/sequence record	macros/execute line
macros/sequence load	

### 1.151 macros/session load

macros/session load of: MACRO MENU

Reads a session file saved by macros/session save : Text buffers and windows are restored according to the contents of the session file. All open text buffers are closed before the old session is restored.

---



```

/* ----- INSERT YOUR CODE HERE: ----- */
'REQUEST BODY="Hi, I''m an empty macro"'
/* ----- END OF YOUR CODE ----- */
'UNLOCK' /* VERY important: unlock GUI */
EXIT

SYNTAX:

SAY "Sorry, error line" SIGL ":" ERRORTXT(RC) ":-("
'UNLOCK'
EXIT

```

## 1.155 macros/macro execute

macros/macro execute of: MACRO MENU

Execute a macro file. ARexx macro examples are placed in the "GoldED:ARexx" drawer. This kind of macro execution should be used if you don't intend to assign the macro to a menu or a key (see event definition).

## 1.156 macros/sequence record

macros/sequence record of: MACRO MENU

Sequences

Start (1st call) respectively stop (2nd call) recording of a sequence. This command usually is assigned to the SHIFT-F10 key, too. During recording all key presses and menu selections are recorded. Mouse movements and mouse clicks are not recorded. Use macros/sequence play to replay a recorded sequence. Use macros/sequence save to save the sequence to disk or macros/sequence load to load a sequence. Saved sequences can be assigned to keys and menus (event definition). If a requester appears while recording you'll often be asked whether this requester should pop up in playback mode, too, or if the selections you made in the requester should be recorded to the macro directly.

## 1.157 macros/sequence load

macros/sequence load of: MACRO MENU

Loads a recorded sequence macros/sequence record .

## 1.158 macros/sequence save

macros/sequence save of: MACRO MENU

Saves recorded sequence, created by macros/sequence record . Sequences should be saved to golded:recordings, the file extension should be "\*.seq".

## 1.159 macros/sequence play

macros/sequence play of: MACRO MENU

Replays recorded sequence (see macros/sequence record ).

## 1.160 macros/sequence play...

macros/sequence play... of: MACRO MENU

Replays recorded sequence (see macros/sequence record ) multiple times. Sequence playback stops if an error occurs (e.g. if the find function can't find the search pattern).

## 1.161 macros/macros

macros/macros of: MACRO MENU

MACRO REVISION (by Marius Gröger)

Updates version string, revises history: This function scans your text for a version string matching the format described in Commodore's programmers' style guide. The revision number is increased by one if a version string is found. Versions strings are embedded into programs to provide required information for the AmigaDOS VERSION command. The following format is supported:

```
$VER <program name> <version number>.<revision number> (<date>)
```

Additionally, you are asked for a short comment about the new revision if a '\$HISTORY:' section has been detected within the text. The comment is added at the top of the history list. Example text header understood by the REVISION macro:

```
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
*/
```

#### MACRO NUMBER LINES

Numbers a text. You are asked for the start value and the step value to be used. Choosing 1000 and 10 would start numbering at 1000 (1000, 1010 ...).

#### MACRO FOLD

Folds marked lines (see block/mark on how to mark lines). You are asked for a comment to be used as fold header. Please read the Folding section of this manual if you are unfamiliar with GoldED's folding feature.

#### MACRO C FUNCTION

Inserts an empty function body (C style) into the text. You are asked for the function's name (e.g. "main") and the return type (e.g. UWORD). This macro accepts two BOOL arguments (TRUE, FALSE): one controls if a comment should be inserted before the function, the other controls if fold markers should be inserted, too. Default is that only a comment is inserted. Edit the menu to modify the defaults ( extras/options local/menu ).

#### MACRO C BLOCK

Marks all lines between the nearest pair of curly brackets.

## 1.162 macros/execute command

### macros/execute command

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(): same syntax as used by most DOS commands (e.g. put strings into quotation marks). Example: INFO VERSION

## 1.163 macros/execute line

### macros/execute line

Executes the current line as DOS command. Useful to embed compiler calls (hidden as comments) into source codes. Included below is an example file header demonstrating how to embed the DICE-C compile command into a source code:

```
/* -----
ED v3.0 - GoldED quick starter, ©1996 Dietmar Eilert. DICE:
```

```
dcc main.c -// -proto -mRR -mi -r -2.0 -o ram:ED
```

```
-----  
*/
```

## 1.164 About GoldED

About GoldED

Displays version information. Additionally, the name of GoldED's ARexx port and the screen name used by the editor are displayed. Screen created by GoldED always are public, i.e. you are invited to run other tasks on the same screen. Example:

```
SHELL CON:0/11/640/100/Shell/screenGOLDED.1
```

... opens a shell window on GoldED's screen (note that there is no space between the "screen" keyword and the screen's name "GOLDED.1").

## 1.165 Keyboard

Keyboard

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

Please read this chapter carefully if you want to take full advantage of this editor's features. Several useful functions assigned to keys are not available in the menu. Other functions are available in the menu, too, but processing of menus (and menu shortcuts) is considerably slower than processing keys.

Keys usually perform different tasks depending on what qualifier key(s) are pressed simultaneously. Qualifier keys are SHIFT, ALT or CTRL. For example, the cursor keys offer seven different functions depending on what qualifiers are used. Key bindings are not fixed; use extras/options local/keyboard to adjust them to your likings. This manual describes the default setup.

## 1.166 Cursor keys

Cursor keys

CURSOR UP/DOWN + ALT

This sequence provides fast scrolling (up or down). Cursor position doesn't

---

change (e.g. the cursor will stay in the middle of the screen if it was there before you started scrolling).

#### CURSOR UP/DOWN + SHIFT

Moves the cursor to the next (DOWN) or the previous (UP) page. Pages do overlap to make navigation more comfortable.

#### CURSOR 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.

#### CURSOR LEFT/RIGHT + ALT

Shifts the display area to the left or right (without moving the cursor). Usually the display is shifted automatically if the cursor reaches the right/left window borders.

#### LEFT/RIGHT + CTRL

Indents marked block ( block/mark ). Hold down the SHIFT key simultaneously to indent one tab position instead of one character.

#### LEFT/RIGHT + SHIFT

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

#### LEFT/RIGHT + SHIFT + ALT

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

## 1.167 HELP key

### HELP key

#### HELP

Folds/unfolds 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.

---

## 1.168 TAB key

TAB key

TAB (+ SHIFT)

Moves the cursor to next tab position. This editor supports regular tabs (e.g. every 4th column), fixed tabs (any column you want) and dynamic tabs. Use extras/options local/tabs to set mode & tab positions. Press the SHIFT key simultaneously if you want to jump to the previous instead of to 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).

TAB + ALT (+ SHIFT)

Holding down ALT while you press the tab key instructs GoldED to use fixed tabs no matter what tab mode has been selected.

## 1.169 RETURN key

RETURN key

RETURN

Splits current line at cursor position and moves the cursor to the next line. This editor supports AutoIndentation : if you press the RETURN key, the current line's indentation is used as default indentation for the next line. Additionally, context sensitive indentation (SmartIndentation] is available: indentation depends on the text.

RETURN + SHIFT

Similar to RETURN except that the cursor line is not splitted at the cursor position.

RETURN + CTRL

Inserts an empty line before the cursor line.

## 1.170 DEL key

DEL key

DELETE (+ SHIFT)

Deletes character under the cursor and shifts the rest of line one position to the left. Deletes until the end of line if the SHIFT key is held down simultaneously.

DELETE + CTRL

---

Deletes the current line. You can recall up to 50 deleted lines using extras/line insert command. Similar to extras/line remove but faster (no menu handling required).

DELETE + ALT

Deletes the current word. You can recall up to 100 deleted words (DEL-ALT-SHIFT).

DELETE + ALT + SHIFT

Reinsert deleted word (see above).

## 1.171 ESC key

ESC key

AutoCompletion

Completes the word under the cursor based on the user dictionary ( extras/options local/words ). Example usage: type 'TIG', then press the ESC key. 'TIG' is replaced by 'TAG\_IGNORE' if the C-dictionary is present. GoldED uses a pattern matching algorithm to find appropriate dictionary entries: you can use other abbreviations than 'TIG', too (e.g. 'TAGI'). However, the first letter of the word to be completed must match the first letter of the full form. The number of letters required to ensure unique identification depends on the dictionary size (keep your dictionary small). Try GoldED's IntelliSense function if you don't like the idea of maintaining a dictionary.

## 1.172 SPACE key

SPACE key

IntelliSense

One major disadvantage of GoldED's dictionary based completion function ( APC ) is that a dictionary has to be maintained. The IntelliSense feature doesn't suffer from this restriction: it completes words based on the context, scanning the document to find a possible expanded form. Type the first few letters of the word, then call the IntelliSense function (assigned to SHIFT-SPACE): the word is completed. The number of letters to be typed depends on the context and requires some experience.

Technical information: Strategy and speed of the IntelliSense function are controlled by a "trigger parameter" and can be adjusted if you edit the menu/keyboard configuration. The trigger parameter determines how thoroughly the context is searched. Default is 4. Useful values range from 1 to 10. The smaller, the faster.

---

## 1.173 F-Keys

### F-Keys

Function keys are configurable (as are all other keys): use the extras/options local/keyboard requester to configure the keyboard. However, some commodities consume all F-keys (including the F-key commodity): GoldED will not recognize or handle function keys if such a commodity is installed. Default mappings of the function keys are:

key	description	see
-----		
F1	open file	project/open
F2	merge file	project/insert
F3	print file	project/print
F4	hide block	block/mark hide
F5	mark line	block/mark
F6	find next	search/find next
F7	next page	cursor keys
F8	set insert mode	layout/insert mode
F9	toggle tab mode	layout/solid tabs
F10	play macro	macros/sequence play
SHIFT		
& key	description	see
-----		
F1	save as	project/save as
F2	save as XPK	project/save as XPK
F3	quit	project/quit buffer
F4	clear text	project/clear window
F5	mark line	block/mark
F6	replace	search/replace
F7	page up	cursor keys
F8	overwrite mode	layout/insert mode
F9	project setup	extras/options local/project
F10	record macro	macros/sequence record

## 1.174 ARexx port

### ARexx port

#### ARexx macros vs. recorded sequences

GoldED supports two methods of automated control: ARexx scripts and recorded sequences. ARexx scripts are programs similar to programming languages like BASIC. They are started by GoldED (examples: macros/macros ) but are actually executed by the ARexx server REXXMast (part of the Amiga operating system, usually started in the startup-sequence). The ARexx server will run the script line-by-line and send messages to GoldED if it detects commands it doesn't understand (commands intended to be handled by GoldED). ARexx is a very powerful method to automate control. 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/sequence record 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/sequence play ). You may save sequences ( macros/sequence save ) or assign them to events like menus or keystrokes. Sequences are far less powerful than ARexx script. But they are more handy.

The following sections describe the editor's ARexx interface. We expect that you are familiar with ARexx basics, i.e. you should know about the purpose of ARexx, how to write scripts, how to talk to applications, etc.

## ARexx basics

It is important that ARexx macros made for GoldED follow a special protocol to avoid collisions with user input. The protocol consists of the following steps:

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

### 1.175 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/macro start ), any script commands which are not part of ARexx itself are sent to the editor (the 'host') automatically. However, if your script is started by a different program (e.g. rx), it will have to select a port (GoldED's port) for communication on its own: 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 port of a further editor task would be "GOLDED.2". Select about GoldED (or use the QUERY command ( macros/execute command : QUERY HOST) if you want to know the current port name and screen name. Most users will never run more than one GoldED task, so assuming that the port is "GOLDED.1" should work in most cases.

### 1.176 Lock a window

Lock a window

Your script will have to tell the editor what window is used first: send the LOCK command (e.g. LOCK CURRENT RELEASE=4 to lock the current window). This command locks the complete user interface: the user will not be able interfere with your macro. The return code has to be checked: RC=0 signals

---

that GoldED has been locked successfully. Take care of unlocking the GUI before your script exits: the user will not be able to use the editor if a script exits without unlocking the GUI. Programming errors in macros could leave the editor in a locked state: use error trapping commands provided by ARexx to solve this problem. In interactive mode, results of the QUERY command are displayed in a window (e.g. LINE=3). In ARexx mode (after a LOCK ), results are not displayed but written to the variable RESULT without prefixed variable name (e.g. 3).

## 1.177 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, then by GoldED using the ReadArgs() function of the DOS library. This sometimes screws things up (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 returns command results to your script using the special ARexx variable RESULT (provided that you have asked for results using the ARexx command OPTIONS RESULTS). Some commands (e.g. QUERY ) allow you to specify a variable to receive the result (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 is 0 if a command has been processed successfully). Additional error explanations usually are available in the special variable RC2.

## 1.178 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 script examples in the golded:arexx drawer. As a last resort the Unlock macro (GoldED drawer) is provided: doubleclick at its icon to unlock all editors.

## 1.179 Internal commands

### Internal commands

GoldED offers a set of about 250 commands and options (see Command list ), supported by all interfaces of GoldED: you may use them in ARexx macros, bind them to menu items (see extras/options local/menu ), attach them to keys ( extras/options local/keyboard ) or execute them directly using macros/execute command . DOS rules apply as far as arguments are concerned since GoldED uses the ReadArgs() function of the Amiga operating system: strings containing spaces must be quoted, option and keywords may be uppercase or lowercase. The line below is an example for the syntax descriptions found on the following pages:

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

The PRINT command obviously accepts five options: FORCE, ITALICS, ALL, LPI and CONFIG. Usage of these options depends on the option type, indicated by a single letter after the slash: some options are used to pass numbers to GoldED, other introduce strings, etc. All options can be used simultaneously. Options followed by "/A" may not be omitted.

The option FORCE is a switch (/S = "switch"): it makes the print command behave in a special way (described in this manual) if this option is specified. No further arguments are expected.

```
PRINT ALL
```

The second option ITALICS introduces a keyword/value pair (/K = "key"): an argument is expected after the option name. The argument type depends on the context (the command) and is described in this manual: some commands expect a text argument, other commands expect boolean values (TRUE or FALSE). Example:

```
PRINT ALL ITALICS=TRUE
```

The equality sign may be omitted. Quotation marks can not be used directly in text arguments: the editor would misinterpret them as "end of string" marker. Put a star in front of the quotation mark (\*) to tell the parser that the next quotation mark doesn't mark the end of a string (single "\*" have to be written as "\*\*"). The LPI option in the example above introduces a numerical (/N) argument. Example:

```
PRINT ALL LPI=10.
```

The equality sign may be omitted. The valid argument range depends on the command. Further option types not shown in the example are "\F" (accepts rest of line as string, no quotation marks required) and "\M" (accepts multiple strings separated by spaces).

## 1.180 Command list

Command list (use: see internal commands ):

---

API	DPAGE	HELP	NAME	REDO	TABS
BACK	ELSE	HUNTER	NEW	REFRESH	TASK
BEEP	ENDIF	IF	NEXT	REMAP	TEXT
BIND	ENDWORD	INDENT	NOTIFY	REPLACE	TMPLATE
BITS	EXALL	INFO	OPEN	REQLIST	TYPE
BLOCK	EXTRACT	INSERT	PATH	REQUEST	UJUMP
BRACKET	FDOWN	KEY	PC	RIGHT	UNDO
BREAKPT	FILE	LAYOUT	PHRASE	RUN	UNLOCK
CLIP	FIND	LEFT	PING	RX	UP
CMD	FIRST	LINES	PONG	SAVE	UPAGE
CODE	FIX	LOCK	POP	SCREEN	USE
COLON	FOLD	MACRO	PREFS	SESSION	VIEW
CR	FORMAT	MARK	PREV	SET	VLEFT
DEBUG	FREEZE	MAXDOWN	PREVEND	SHIFT	VRIGHT
DEL	FUNC	MAXUP	PRINT	SMARTCR	WINDOW
DELETE	FUP	MENUS	PROJECT	STOP	WORD
DIR	GOTO	MISC	PUSH	SUFFIX	XREF
DJUMP	GREP	MORE	QUERY	SYNTAX	
DOWN	GUI	MOUSE	QUIT	TAB	

## 1.181 API

API

Description of internal commands

command	option	description
API	ASK/S	open API configuration requester

## 1.182 BACK

BACK

Description of internal commands

command	option	description
BACK	(no options)	backspace operation
	SMART/S	backspace over marked word deletes word

## 1.183 BEEP

BEEP

Description of internal commands

---

command	option	description
-----		
BEEP	(no options)	beep

## 1.184 BIND

BIND

Description of internal commands

command	option	description
-----		
BIND	ASK/S	open keyboard configuration requester

## 1.185 BITS

BITS

Description of internal commands

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

## 1.186 BLOCK

BLOCK

Description of internal commands

command	option	description
-----		
BLOCK	UPPER/S	convert block to uppercase
	LOWER/S	convert block to lowercase
	SORT/S	sort block
	CHKCASE/S	sort block: consider case
	CURSOR/S	sort option: compare starts at cursor position
	COPY/S	copy block to cursor position
	MOVE/S	move block to cursor position
	HIDE/S	hide marker after operation

## 1.187 BRACKET

### BRACKET

Description of internal commands

command	option	description
BRACKET	MATCH/S	move cursor to matching bracket
	CHECK/S	check use of () in current line
	TWINS/K	bracket type to be considered (STRING, e.g. "()")

## 1.188 BREAKPT

### BREAKPT

Description of internal commands

command	option	description
BREAKPT	LINE/N	line number (ULONG: 1, ...)
	LOCKED/S	use locked debugger line numbers
	UNFOLD/S	unfold
	STATE/N	new breakpoint state (UWORD: 0, 1)
	TOGGLE/S	toggle breakpoint state
	NEXT/S	move cursor to next breakpoint
	PREV/S	move cursor to prev breakpoint
	SAVE/S	save breakpoints
	ALL/N	set breakpoints (UWORD: 0, 1)
	CLEANUP/S	delete all breakpoints in all buffers
	FILE/K	file name
	FORMAT/K	debugger data format (STRING)
	DATA/N	debugger data (APTR)
	ELEMENTS/N	debugger data size (ULONG)
	NONOTIFY/S	don't notify debugger
	QUERY/K	breakpoint array request (STRING)

Comment: The options FILE/K, FORMAT/K, DATA/N, ELEMENTS/N, NONOTIFY/S and QUERY/K are reserved for use by debuggers.

## 1.189 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

VPASTE/S vertical clipboard paste  
 UNIT/N clipboard unit to use (UBYTE); defaults is 0

## 1.190 CMD

CMD

Description of internal commands

command	option	description
CMD	(no options)	open command requester

## 1.191 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	toggle case of character under cursor
	UPPER/S	convert character under cursor to uppercase
	LOWER/S	convert character under cursor to lowercase

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.

## 1.192 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.

## 1.193 CR

CR

Description of internal commands

command	option	description
CR	(no options)	<RETURN> command; splits line at cursor position
		Comment: see extras/options local/indent

## 1.194 DEBUG

DEBUG

Description of internal commands

command	option	description
DEBUG	START/S	start debugger support
	STOP/S	stop debugger support
	PORT/K	ARexx port of debugger (STRING)
		Comment: Reserved for usage by external debugger.

## 1.195 DEL

DEL

Description of internal commands

command	option	description
DEL	(no options)	deletes character under cursor

## 1.196 DELETE

DELETE

Description of internal commands

command	option	description
DELETE	WORD/S	delete next word
	EOW/S	delete until end of word
	SMART/S	consider white space settings
	EOL/S	delete until end of line

LEFT/S delete until beginning 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

Comment: The last 100 deleted words (WORD/S) can be reinserted using INSERT (INSERT WORD).

## 1.197 DIR

DIR

Description of internal commands

command	option	description
DIR	ASK/S	open requester to set current directory
	NEW/F	set current directory (STRING)
	CURRENT/S	use document's path as new current directory

Comment: See project/current dir

## 1.198 DJUMP

DJUMP

Description of internal commands

command	option	description
DJUMP	(no options)	jump to bottom of window/next page

Comment: Cursor jumps to the window's last line if placed above that line so far. Jumps to the next page otherwise. Compare: DPAGE .

## 1.199 DOWN

DOWN

Description of internal commands

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

## 1.200 DPAGE

DPAGE

Description of internal commands

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

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

## 1.201 ELSE

ELSE

Description of internal commands

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

else	(no options)	introduces alternative IF section
------	--------------	-----------------------------------

## 1.202 ENDIF

ENDIF

Description of internal commands

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

ENDIF	(no options)	terminates IF
-------	--------------	---------------

## 1.203 ENDWORD

ENDWORD

Description of internal commands

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

ENDWORD	(no options)	move cursor to end of word
---------	--------------	----------------------------

## 1.204 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).

**1.205 EXTRACT**

## EXTRACT

Description of internal commands

command	option	description
EXTRACT	(no options)	extract file name under cursor
	VAR/K	where to put the result: ARexx variable (STRING)
	LEFT/K	left delimiter(s) (STRING); e.g. "<[("
	RIGHT/K	right delimiter(s) (STRING); e.g. ">]"

Comment: Left & right delimiter strings must be of paired and of the same length. Priority is from left to right.

**1.206 FDOWN**

## FDOWN

Description of internal commands

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

**1.207 FILE**

## FILE

Description of internal commands

command	option	description
FILE	NAME/K	file to deleted/searched (STRING)
	DELETE/S	delete file
	RENAME/S	rename file

FORCE/S            don't ask for confirmation  
 SEARCH/K          search this path for specified file (STRING)  
 VAR/K             where to put the result: ARexx variable (STRING)  
 NEWDIR/K          create directory (STRING)

Comment: Delete-protected files are not deleted unless the FORCE mode is used.

## 1.208 FIND

FIND

Description of internal commands

command	option	description
FIND	STRING/K	search pattern (STRING)
	BLOCK/S	search marked text
	INDEX/S	list found strings in a requester
	WILD/K	set wildcard mode (BOOL)
	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')
	WORDS/K	look for whole words only ? (BOOL)

## 1.209 FIRST

FIRST

Description of internal commands

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

## 1.210 FIX

FIX

Description of internal commands

command	option	description
FIX	VAR/K/A	ARexx variable name (STRING)

Comment: To be used within macros only. Fixes the contents of the given ARExx string variable to make it "parser-proof" (e.g. handles embedded '"'); compare internal commands ).

## 1.211 FOLD

FOLD

Description of internal commands

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

## 1.212 FORMAT

FORMAT

Description of internal commands

command	option	description
FORMAT	LINES/S	format paragraph
	MARK/S	format block
	LEFT/S	mode: left -aligned
	RIGHT/S	mode: right-aligned
	BLOCK/S	mode: justified
	CENTER/S	mode: centered

## 1.213 FREEZE

FREEZE

Description of internal commands

command	option	description
FREEZE	CURRENT/S	hide current window
	ASK/S	show list of hidden buffers
	SWAP/S	swap current/next frozen window
	PREV/S	swap backwards
	ADD/M	load file(s) directly to frozen list

## 1.214 FUNC

## FUNC

Description of internal commands

command	option	description
FUNC	(no options)	open Quickref requester
	MODE/K	mode selection: scanner name (STRING)
	SMART/S	mode selection: automatic according to file name
	INDEX/S	mode selection: standard search
	CURRENT/S	search reference for word under cursor
	UNFOLD/K	examine folds (BOOL)
	SORT/S	sort list

**1.215 FUP**

## FUP

Description of internal commands

command	option	description
FUP	(no options)	scroll backwards (fast)

**1.216 GOTO**

## GOTO

Description of internal commands

command	option	description
GOTO	LINE/N	line number to go to (ULONG: 1, ...)
	COLUMN/N	column to go to (UWORD: 1, ...); see FIRST
	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	move cursor after last character of line
	INDENT/S	move cursor to first character of line
	BFIRST/S	go to beginning of block
	BLAST/S	go to end of 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
	BYTE/N	byte offset to go to (LF's included)
	LOCKED/S	use locked debugger line numbers

Comment: Line numbers are expected to be absolute numbers if UNFOLD=TRUE is set. Folded blocks count as a single line in UNFOLD=FALSE mode.

## 1.217 GREP

GREP

Description of internal commands

command	option	description
GREP	STRING/K ASK/S case/K	string to search project files for (STRING) ask for string to search for case sensitive search ? (BOOL)

## 1.218 GUI

GUI

Description of internal commands

command	option	description
GUI	ASK/S	open GUI configuration window

## 1.219 HELP

HELP

Description of internal commands

command	option	description
HELP	CATALOG/K TOPIC/K STOP/S	guide file (STRING) node to be displayed (STRING, e.g. "MAIN") stop guide task

Comment: The menu's guide is used ( extras/options local/menu ) if no guide is specified.

## 1.220 HUNTER

HUNTER

Description of internal commands

---

command	option	description
HUNTER	ASK/S	open hunter configuration window
	CURRENT/S	hunt (i.e. search & open) filename under cursor
	NAME/K	search this file (STRING)
	DEEP/K	scan subdirectories ? (BOOL)

## 1.221 IF

IF

Description of internal commands

command	option	description
IF	VAR/K	internal variable (STRING); see Variables
	MATCH/K	pattern (STRING)
	NOT/S	negate result

Comment: Disables menu/mouse/keyboard related command execution until the next ENDIF if the contents of VAR/K doesn't match the pattern.

## 1.222 INDENT

INDENT

Description of internal commands

command	option	description
INDENT	ASK/S	open indentation configuration requester

## 1.223 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

## 1.224 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	BLOCK/S: insert empty column; see AT/N
	STRING/K	BLOCK/S: insert text; see AT/N (STRING)
	AT/N	BLOCK/S: column where to insert (UWORD)
	APPEND/S	BLOCK/S: append text to marked lines
	WORD/S	reinsert deleted word (see DELETE )
	PATH/S	insert filename (file requester selection)

## 1.225 KEY

### KEY

Description of internal commands

command	option	description
KEY	EVENT/K	input event description (STRING)
	RAW/S	event is a plain character sequence

Comment: See input events for details

## 1.226 LAYOUT

### LAYOUT

Description of internal commands

command	option	description
LAYOUT	ASK/S	open layout configuration requester
	LEFT/S	use cursor position as left border
	RIGHT/S	use cursor position as right border

Comment: Borders can be set using SET , too.

## 1.227 LEFT

### LEFT

Description of internal commands

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

## 1.228 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	duplicate current line

## 1.229 LOCK

### LOCK

Description of internal commands

command	option	description
LOCK	CURRENT/S	lock current window
	NAME/K	lock this window (STRING: file name)
	QUIET/S	don't activate window
	RELEASE/N	required editor version (ULONG, this release: 4)

Comment: To be used within ARexx macros. Return code has to be checked. An UNLOCK command must follow later to avoid deadlocks. LOCK can not be nested: a single UNLOCK unlocks all locks. This call moves GoldED's screen to the front.

## 1.230 MACRO

### MACRO

Description of internal commands

command	option	description
MACRO	RECORD/S	start/stop sequence recording
	PLAY/S	play 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 recorded sequence  
LOAD/S            load a sequence

Comment: Sequence playback is disabled during execution of  
ARexx macros.

## 1.231 MARK

### MARK

Description of internal commands

command	option	description
MARK	HIDE/S	hide mark
	PARAGRAPH/S	mark paragraph
	WORD/S	mark word under cursor
	SET/S	set marker start/end
	BEGIN/S	set beginning of block
	END/S	set end of block
	FLOW/K	should cursor movements size block ? (BOOL)
	EXCLUDE/K	cursor part of block ? (BOOL)
	LINE/S	resolution = lines
	COLUMN/S	resolution = characters
	STRICT/S	only SPC (ASCII 32) is regarded as word delimiter

## 1.232 MAXDOWN

### MAXDOWN

Description of internal commands

command	option	description
MAXDOWN	(no options)	move cursor to next quarter of document

## 1.233 MAXUP

### MAXUP

Description of internal commands

command	option	description
MAXUP	(no options)	move cursor to previous quarter of document

## 1.234 MENUS

### MENUS

Description of internal commands

command	option	description
MENUS	ASK/S	open menu configuration requester

## 1.235 MISC

### MISC

Description of internal commands

command	option	description
MISC	ASK/S	open misc configuration requester

## 1.236 MORE

### MORE

Description of internal commands

command	option	description
MORE	(no options)	open further window
	NAME/K	window name (STRING); default: "unnamed"
	FILETYPE/K	overrides automatic file type selection (STRING)
	SMART/S	use current window if empty

## 1.237 MOUSE

### MOUSE

Description of internal commands

command	option	description
MOUSE	ASK/S	open mouse preferences requester
	SET/S	move cursor to mouse position
	MARK/S	mark block (to be used in conjunction with SET/S)
	LINE/S	mark whole lines only
	DRAG/S	activate drag & drop support

Comment: SET/S, MARK/S, DRAG/S and LINE/S are reserved for mouse configuration.

## 1.238 NAME

NAME

Description of internal commands

command	option	description
NAME	ASK/S	display text name requester
	NEW/F	set new text name (STRING)

## 1.239 NEW

NEW

Description of internal commands

command	option	description
NEW	(no option)	clear text (user is asked for confirmation)
	FORCE/S	no user confirmation
	NONAME/S	reset name to 'unnamed'

## 1.240 NEXT

NEXT

Description of internal commands

command	option	description
NEXT	(no options)	move cursor to next word

## 1.241 NOTIFY

NOTIFY

Description of internal commands

command	option	description
NOTIFY	FILE/K/A	file to be monitored (STRING)
	START/S	start monitoring

STOP/S            stop monitoring  
 CHECK/S         number of write accesses since last check (UWORD)  
 MACRO/K         macro to be executed upon write access (STRING)

Comment: Provides access to the notification mechanism of AmigaDOS. The file name is passed as argument to the macro.

## 1.242 OPEN

OPEN

Description of internal commands

command	option	description
OPEN	NAME/M	file(s) to open (STRING or STRINGS)
	FAST/S	fast loading
	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	display file requester
	QUIET/S	don't complain about missing files
	PATH/K	default path for file requester (STRING)
	OLDPATH/S	use path of current text as default path
	SMART/S	use current window if empty
	RAW/S	don't expand tabs to spaces
	FORCE/S	don't display "file modified" warning
	FILETYPE/K	overrides automatic file type selection (STRING)

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

## 1.243 PATH

PATH

Description of internal commands

command	option	description
PATH	ASK/S	open hunter configuration requester

## 1.244 PC

PC

Description of internal commands

---

command	option	description
PC	COLUMN/N	column (UWORD: 1, ...)
	LINE/N/A	line (UWORD: 1, ...)
	FILE/K	buffer name/file name (STRING)
	PRESENT/S	activate window
	HIGHLIGHT/S	highlight line
	LOCKED/S	use locked debugger line numbers
	UNFOLD/S	unfold
	ONEWINDOW/S	don't use multiple windows while debugging
	FREEZE/S	hide unused buffers (default: unload buffers)
	FILETYPE/K	overrides filetype detection (STRING)
	FORMAT/K	debugger data format (STRING)
	DATA/N	debugger data (APTR)
	ELEMENTS/N	debugger data size (ULONG)

Comment: Reserved for usage by debugger.

## 1.245 PHRASE

### PHRASE

Description of internal commands

command	option	description
PHRASE	ASK/S	open dictionary configuration requester
	CURRENT/S	complete current word
	SMART/S	activate IntelliSense completion
	ADD/K	add keyword to dictionary (STRING)
	TRIGGER/N	IntelliSense trigger parameter (UWORD: 1...10)

## 1.246 PING

### PING

Description of internal commands

command	option	description
PING	SLOT/N	save cursor position to slot (UWORD: 0...9)

Comment: see PONG

## 1.247 PONG

### PONG

Description of internal commands

command	option	description
PONG	SLOT/A/N	recall cursor positions (UWORD 0-9)
Comment: GoldED offers ten bookmark storages (0-9) for each window. Slot 0 usually is reserved for use within ARExx scripts (see PING ).		

## 1.248 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.		

## 1.249 PREFS

PREFS

Description of internal commands

command	option	description
PREFS	ASK/S	open configuration requester
	LOCAL/S	local configuration
	GLOBAL/S	global configuration
	SAVE/S	save configuration
	PAGE/N	preselect notebook page (UWORD: 0, ...)

## 1.250 PREV

PREV

Description of internal commands

command	option	description
PREV	(no options)	move cursor to previous word

## 1.251 PREVEND

PREVEND

Description of internal commands

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

## 1.252 PRINT

PRINT

Description of internal commands

command	option	description
PRINT	ASK/S	open printer configuration requester
	FORCE/S	no "print ?" confirmation requester
	WINDOW/S	print hardcopy of window
	ALL/S	text printing: print complete file
	BLOCK/S	text printing: print block

## 1.253 PROJECT

PROJECT

Description of internal commands

command	option	description
PROJECT	ASK/S	open project configuration requester
	ADD/K	add source file (STRING)
	DEL/N	remove a source file (UWORD: 0, ...)
	CLR/S	clear project list
	LIST/N	copy this list (struct List *)

## 1.254 PUSH

PUSH

Description of internal commands

command	option	description
PUSH	(no options)	insert last line of push / pop buffer into text
	KEEP/S	do not delete line from buffer

## 1.255 QUERY

### QUERY

Description of internal commands

command	option	description
QUERY	NAME/A	variable name (STRING)
	VAR/K	variable to hold result (STRING); default: RESULT

Comment: Returns value of the specified internal variable. This function may be used in macros and in interactive mode (see macros/execute command). In interactive mode a requester displays the result. In non-interactive mode the result is stored in a variable. You may specify the name of an ARExx variable (VAR/K, defaults to RESULT) or of an environment variable (first letter of variable name = '\$') to hold the result. Valid variable names are:

### Variables

Name	Type	Description
ABAK		BOOL AutoBackups enabled ?
ABSLINE	READONLY	NUMBER absolute line number; see LINE
ABSLINES	READONLY	NUMBER absolute lines; see LINES
ABSNAME		STRING absolute file name (expanded); see DOC
ACENTER		BOOL center new windows ?
AFOLD		BOOL AutoFold after loading ?
ALEFT		BOOL layout: use current indentation ?
ALT	READONLY	BOOL ALT key pressed ?
ANSI	READONLY	NUMBER number of non-ASCII-characters (*)
ANYCHAR	READONLY	BOOL current line not empty ?
ANYFOLDS	READONLY	BOOL any folds in document ?
ANYTEXT	READONLY	BOOL current window not empty ?
ARRANGE		BOOL AutoArrange windows ?
ASKBAK		BOOL confirm backups ?
AUTOINDENT		BOOL keep indentation after CR ?
BACKUP		BOOL create backups ?
BAKDIR		STRING backup directory
BLOCK	READONLY	NUMBER block type (0=hidden 1=lines 2=characters)
BLOCKB	READONLY	NUMBER block end (line)
BLOCKR	READONLY	NUMBER block end (column)
BLOCKX	READONLY	NUMBER block start (column)
BLOCKY	READONLY	NUMBER block start (line)
BOLD		BOOL bold printing ?
BRACKET		BOOL automatic () checks ?
BUFFER	READONLY	STRING contents of current line
BYTES	READONLY	NUMBER text size (bytes) (*)
CAPS	READONLY	BOOL CAPSLOCK key pressed ?
CAT	READONLY	STRING catalog used by GoldED
CHKCASE		BOOL AutoCase enabled ?
CODE	READONLY	NUMBER ASCII code of character under cursor
COLUMN	READONLY	NUMBER cursor column (1, ...)
COLUMNS	READONLY	NUMBER window width (1, ...); see WINW

CON	READONLY	STRING	recommended console description string
CTRL	READONLY	BOOL	CTRL key pressed ?
CURRENT	READONLY	NUMBER	input buffer (char *)
DATE		STRING	date
DEBUG		BOOL	save breakpoints ?
DEBUGGER		BOOL	breakpoint display visible ?
DOC		STRING	document's name, path included (see ABSNAME)
DOUBLE		BOOL	double printing ?
EOL		BOOL	EOL wrap ?
ERR	READONLY	STRING	last error message
EXCLUDE		BOOL	cursor inside/outside marker while marking
FILE		STRING	file name (without path); see DOC
FIND		STRING	search pattern
FOLDA		STRING	fold marker: start
FOLDB		STRING	fold marker: end
FOLDS	READONLY	NUMBER	number of folds in text (*)
FUNC	READONLY	STRING	QuickFunc scanner
HANDLE	READONLY	NUMBER	internal window handle
HMI		NUMBER	printer line feed (0 - 2)
HOST	READONLY	STRING	ARexx port name
INBLOCK	READONLY	BOOL	cursor in block ?
INFO		STRING	icon file (*.info)
INFOS		BOOL	create icons ?
INSERT		BOOL	insert mode ?
ITALICS		BOOL	italics printing ?
LEFT		NUMBER	layout: left border
LEN	READONLY	NUMBER	length of current line
LEVEL	READONLY	NUMBER	syntax level of character under cursor
LINE	READONLY	NUMBER	line number (relative); ABSLINE
LINES	READONLY	NUMBER	lines (relative); see ABSLINES
LPI		NUMBER	lines/inch: 6 lpi or 8 lpi (UWORD: 0, 1)
LQ		BOOL	letter quality printing ?
MARKED	READONLY	BOOL	block marked ? See BLOCK
MAXLEN	READONLY	NUMBER	length of longest line (*)
MODIFY	READONLY	BOOL	text modified ?
NODEMO	READONLY	BOOL	registered version ?
NUMPAD		BOOL	numpad configurable ?
ORDINAL	READONLY	NUMBER	window number (0, ...)
OUTPUT		STRING	output console
OVERWRITE		BOOL	overwrite files ?
PARSER	READONLY	STRING	syntax parser name
PASSWORD		STRING	password
PATH		STRING	document's path; see DOC
PICKED	READONLY	NUMBER	number of lines in pick/push buffer
PREVIEW		BOOL	preview active ?
PRJLIST	READONLY	NUMBER	project files (struct List *)
PROG	READONLY	STRING	program name
RC	READONLY	NUMBER	error code of last command
READONLY		BOOL	document write protected ?
REFORMAT		BOOL	reformat during word wrap ?
REM		STRING	file comment
RESET		BOOL	reset printer before output ?
RESULT	READONLY	STRING	result string of last command
RIGHT		NUMBER	layout: right border
ROWS	READONLY	NUMBER	window height; see WINH
RPLC		STRING	replace text
RXDEBUG		BOOL	output ARexx debug information ?

SCREEN	READONLY	STRING	public screen name
SCREENH	READONLY	NUMBER	screen height
SCREENW	READONLY	NUMBER	screen width
SHIFT	READONLY	BOOL	SHIFT key pressed ?
SMARTINDENT		BOOL	context-sensitive indentation ?
SOLID		BOOL	tabs insert spaces ?
SPC		STRING	white space definition string
STDLINE	READONLY	BOOL	current line not write-protected ?
SYNTAX		BOOL	syntax highlighting activated ?
TAB		NUMBER	tab distance (keyboard)
TABFILE		NUMBER	tab distance (import/export)
TABMODE		NUMBER	tab mode (0...2)
TEMPLATES		BOOL	templates activated ?
TIME	READONLY	STRING	time
TOOLBAR		BOOL	toolbar visible ?
TOPLINE	READONLY	NUMBER	number of first line in window (relative)
TYPE		STRING	filetype; example (write access): ".c"
UNFOLD		BOOL	GOTO unfolds text ?
USECASE		BOOL	case-sensitive search ?
USER	READONLY	USER	user name
VER	READONLY	STRING	version string
VERSION	READONLY	NUMBER	version code
WILDCARDS		BOOL	wildcard search ?
WINDOWS	READONLY	NUMBER	open windows
WINH	READONLY	NUMBER	window height (units = pixels)
WINW	READONLY	NUMBER	window width (units = pixels)
WORD	READONLY	STRING	word under cursor
WORDS	READONLY	NUMBER	number of words in text (*)
WRAP		BOOL	wordWrap activated ?
X	READONLY	NUMBER	window position (X)
XPK		STRING	selected XPK compressor
Y	READONLY	NUMBER	window position (Y)

(\*) Variable values are not valid until execution of the EXALL command.

## 1.256 QUIT

### QUIT

Description of internal commands

command	option	description
QUIT	(no option)	close current window
	FORCE/S	don't ask for confirmation
	UNLOAD/S	close window, unload editor (see HotKey )
	ALL/S	close all open text buffers

## 1.257 REDO

## REDO

Description of internal commands

command	option	description
REDO	LAST/S	undo last undo

**1.258 REFRESH**

## REFRESH

Description of internal commands

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

**1.259 REMAP**

## REMAP

Description of internal commands

command	option	description
REMAP	TABLE/K	use this conversion file to remap text (STRING)
	ASK/S	open remap requester

**1.260 REPLACE**

## REPLACE

Description of internal commands

command	option	description
REPLACE	STRING/K	search pattern (STRING)
	BY/K	replace pattern (STRING)
	WILD/K	set wildcard mode (BOOL)
	NEXT/S	replace next occurrence of pattern (*)
	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')
	WORDS/K	look for whole words only ? (BOOL)

CONFIRM/K      ask for confirmation ? (BOOL)

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

## 1.261 REQLIST

### REQLIST

Description of internal commands

command	option	description
REQLIST	ENTRY/M/A	strings to be displayed as listview (STRINGS)
	SORT/S	sort list
	VAR/K	ARexx variable: where to put the result (STRING)

Comment: Presents a listview. Selected string is returned.

## 1.262 REQUEST

### REQUEST

Description of internal commands

command	option	description
REQUEST	HIDE/K	disable requesters (BOOL)
	DEFAULT/K	default if requesters are disabled (UWORD)
	BODY/K	body text, lines separated by ' ' (STRING)
	BUTTON/K	button(s) text, separated by ' ' (STRING)
	TITLE/K	requester title (STRING)
	LONG/S	ask for a number
	MIN/N	lower limit for number (WORD)
	MAX/N	upper limit for number/characters (WORD)
	OLD/K	default value (STRING)
	FILE/S	ask for a file
	SAVE/S	put ASL file requester into SAVE mode
	PATH/K	default path if asking for a file (STRING)
	MASK/K	file requester mask (e.g."#?.c") (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)
	STAY/S	turn off automatic status line refresh
	KEY/S	ask for key (returns code and qualifier)
	PROBLEM/K	error message to be displayed (STRING)

Comment: HIDE/K may be used in macros only; re-enable requesters before before the macro is terminated.

## 1.263 RIGHT

## RIGHT

Description of internal commands

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

**1.264 RUN**

## RUN

Description of internal commands

command	option	description
RUN	CMD/K	run this program (STRING)
	DIR/K	current path (STRING); default: path of text
	LINE/S	execute current line of text
	PRIO/N	priority to be used (WORD: -3...3)
	STACK/N	stack to be used (ULONG)
	ASYNC/S	run program asynchronously
	OUTPUT/K	output (STRING)
	SHANGHAI/N	temporary shanghai time: seconds (UWORD)
	WAITPORT/K	wait for appearance of this port (STRING)
	SECONDS/N	WAITPORT timeout; defaults to 5 seconds (UWORD)

**1.265 RX**

## RX

Description of internal commands

command	option	description
RX	CMD/K	command to be send to PORT/K (STRING)
	SYNC/S	send in synchronous mode (default: asynchronous)
	ASK/S	ask for command
	PORT/K	destination: default is "AREXX" (STRING)
	MACRO/K	macro to execute if PORT is not valid (STRING)

Comment: The macro is called with the command string as first argument. Usage similar to the AmigaDOS command RX if you set the port to AREXX: Commands in '.' are interpreted as commands, commands in "..." are interpreted as macro names.

**1.266 SAVE**

## SAVE

Description of internal commands

command	option	description
SAVE	BLOCK/S	save block only
	ALL/S	save complete file
	BUFFERS/S	all modified buffers are save
	FORCE/S	no "overwrite ?" requester
	SMART/S	don't save unmodified file(s)
	NAME/K	file name (where to save) (STRING)
	EXPORT/S	don't rename buffer
	ASK/S	open save-as requester
	EXIT/S	close window if save operation is successful
	CRUNCH/S	compress file (XPK) while saving it
	NOLF/S	don't save linefeeds after text lines

Comment: NOLF should be used carefully since the editor's line length is limited. Loading a text without linefeeds may cause a line length overflow.

**1.267 SCREEN**

## SCREEN

Description of internal commands

command	option	description
SCREEN	ASK/S	open display mode configuration requester
	ICONIFY/S	iconify
	FRONT/S	move GoldED's screen to the front
	BACK/S	move GoldED's screen to the back
	NOSIZE/S	use standard screen size

**1.268 SESSION**

## SESSION

Description of internal commands

command	option	description
SESSION	CONFIG/K	name of a session file (STRING)
	LOAD/S	load session file
	SAVE/S	save session file
	QUIET/S	save modified buffers without confirmation
	NOSAVE/S	don't save modified text buffers

## 1.269 SET

SET

Description of internal commands

command	option	description
SET	NAME/K VALUE/K	variable to be set (STRING): Variables new value (STRING)

## 1.270 SHIFT

SHIFT

Description of internal commands

command	option	description
SHIFT	LEFT/S RIGHT/S COLUMNS/N TAB/S ASK/S LINE/S	shift to the left shift to the right (indent) indentation: number of columns (UWORD) set shifting distance to tab size open requester (left/right shifting) indent line under cursor (default: block)

## 1.271 SMARTCR

SMARTCR

Description of internal commands

command	option	description
SMARTCR	(no options)	return (line is not splitted at cursor position)

## 1.272 STOP

STOP

Description of internal commands

command	option	description
STOP	(no options)	stop command execution

## 1.273 SUFFIX

### SUFFIX

Description of internal commands

command	option	description
SUFFIX	STRING/K	file name (STRING)
	SUFFIX/K	desired suffix (STRING, e.g. ".c")

## 1.274 SYNTAX

### SYNTAX

Description of internal commands

command	option	description
SYNTAX	ASK/S	open syntax highlighting configuration requester
	UNPARSE/S	reset parser cache for current line
	ALL/S	reset parser cache for all lines

## 1.275 TAB

### TAB

command	option	description
TAB	(no option)	standard tab
	BACK/S	backwards

## 1.276 TABS

### TABS

Description of internal commands

command	option	description
TABS	ASK/S	open tab configuration requester

## 1.277 TASK

## TASK

Description of internal commands

command	option	description
TASK	WAIT/N	wait interval (1/50 sec) (UWORD)
	BENCH/S	run graphics benchmark

**1.278 TEXT**

## TEXT

Description of internal commands

command	option	description
TEXT	T/K	text to be inserted at cursor position (STRING)
	VAR/K	variable to be inserted; see QUERY (STRING)
	STAY/S	don't move cursor while inserting text
	CR/S	append linefeed to text

Comment: Use "\*" within T/K to insert quotation marks. Single "\*" have to have written as "\*\*".

**1.279 TMLATE**

## TMLATE

Description of internal commands

command	option	description
TMLATE	ASK/S	open templates configuration requester
	CHECK/S	check word under cursor

**1.280 TYPE**

## TYPE

Description of internal commands

command	option	description
TYPE	RESET/S	autodetect filetype of current text

## 1.281 UJUMP

UJUMP

Description of internal commands

command	option	description
-----		
UJUMP	(no options)	jump to beginning of window/previous page
<p>Comment: Cursor jumps to the windows's first line if placed below that line so far. Jumps to previous page if placed in line one already. Compare: UPAGE .</p>		

## 1.282 UNDO

UNDO

Description of internal commands

command	option	description
-----		
UNDO	LAST/S	undo last operation
	FLUSH/S	free undo data of current text

## 1.283 UNLOCK

UNLOCK

Description of internal commands

command	option	description
-----		
UNLOCK	(no option)	unlock GUI; to be used after LOCK
	DELAY/S	unlock GUI, delay until exit of GoldED
	STICKY/S	unlock GUI, delay until current window is closed
<p>Comment: The DELAY/STICKY options are reserved for use by external applications (e.g. the QuickStarter). They provide ways and means to synchronize with GoldED.</p>		

## 1.284 UP

UP

Description of internal commands

command	option	description
-----		

UP (no options) move cursor one line up

## 1.285 UPAGE

UPAGE

Description of internal commands

command	option	description
UPAGE	(no options)	show previous page (compare UJUMP )

## 1.286 USE

USE

Description of internal commands

command	option	description
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 length of the current line by poking into the line buffer.

## 1.287 VIEW

VIEW

Description of internal commands

command	option	description
VIEW	LEFT/S	shift view left
	RIGHT/S	shift view right
	COLUMNS/N	distance (UWORD); defaults to 5 columns

Comment: VLEFT and VRIGHT provide a better performance and therefore should be preferred.

## 1.288 VLEFT

### VLEFT

Description of internal commands

command	option	description
VLEFT	(none)	shift view 5 columns left

## 1.289 VRIGHT

### VRIGHT

Description of internal commands

command	option	description
VRIGHT	(none)	shift view 5 columns right

## 1.290 WINDOW

### WINDOW

Description of internal commands

command	option	description
WINDOW	MAX/S	maximize current window
	CENTER/S	center current window on screen
	ARRANGE/N	arrange windows (0: vertical, 1: horizontal)
	ZIP/S	zip window
	USE/K	activate named window/file (STRING)
	FORCE/S	load named file if necessary (see USE/K)
	FILETYPE/K	overrides filetype detection (STRING)
	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
	PREV/S	activate previous window
	RECOVER/S	redraw window
	HANDLE/N	activate window using its handle (ULONG)
	ORDINAL/N	activate 1st , 2nd, ... window (ULONG: 0, ...)
	QUIET/S	NEXT/USE/ORDINAL: leave window in the background
	SNAP/S	use current window's dimensions as default size

Comment: A window handle is returned by the OPEN function and by QUERY (QUERY HANDLE).

## 1.291 WORD

WORD

Description of internal commands

command	option	description
WORD	UPPER/S	convert word under cursor to uppercase
	LOWER/S	convert word under cursor to lowercase

## 1.292 XREF

XREF

Description of internal commands

command	option	description
XREF	CURRENT/S	find/open reference file related to current word
	PHRASE/K	find/open file related to this phrase (STRING)
	ASK/S	ask for topic
	CHECK/S	just determine whether a reference is available
	PROTECT/S	write-protect reference windows ?

## 1.293 Input events

Input events

Input event insertion

GoldED's KEY command may be used to insert "input events" (e.g. keystrokes) into intuition's global input stream. Inserting events makes the current application (the active GoldED window) behave as if the user had performed the described action. A key event description string EVENT/K may consist of plain text or plain text mixed with "event descriptions" in angle brackets (e.g. "<shift>"). You would have to specify the RAW/S option if you want to insert plain text containing angle brackets. Examples:

```
KEY EVENT="hello world"
KEY EVENT="hello world<return>"
KEY EVENT="--->" RAW
```

If you do not specify RAW, event descriptions like "<return>" are not treated as plain text but translated into input events (<return> would act as if the return key were pressed). The following expression outlines the format of description strings: <CLASS QUALIFIER(S) KEY>

A) CLASS may be one of the following (assuming <rawkey> if none is given):

```
rawkey ..... this is a keyboard event
```

rawmouse ..... this is a mouse button event

B) QUALIFIER(s) may be one or more of ...

shift ..... shift  
 control ..... ctrl  
 capslock ..... capslock  
 alt ..... alt  
 lcommand ..... left Amiga  
 rcommand ..... right Amiga  
 numericpad ..... numeric pad  
 leftbutton ..... left mouse button  
 rbutton ..... right mouse button

C) KEY may either be a plain character or ...

space ..... space  
 backspace ..... backspace  
 tab ..... tab  
 enter ..... enter  
 return ..... return  
 esc ..... esc  
 del ..... delete  
 up ..... cursor up  
 down ..... cursor down  
 right ..... cursor right  
 left ..... cursor left  
 f1 - f10 ..... function key  
 help ..... help

Examples: KEY EVENT="`<rawkey shift A>`"  
 KEY EVENT="`<rawkey f1>`"  
 KEY EVENT="`<rawkey shift down>`"  
 KEY EVENT="`<rawkey rcommand o>`"

## 1.294 TOOLS

### TOOLS

GoldED is shipped with a couple of useful tools supporting your daily work with this package; some of them are described on the next pages:

HiSpeed ..... Printer utility for LaserJet and DeskJet printers  
 Recover ..... Recover lost text buffers after a system crash

## 1.295 HiSpeed

Tools : HiSpeed

The HiSpeed printer utility has been designed for PCL printers like Hewlett Packard's DeskJet and LaserJet family. It can be used to reduce the amount of paper when printing large files since it is able to print with small typefaces and to multiple columns on both sides of a sheet. 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. You will like this program if you have to do a lot of printing. HiSpeed is shareware. The registered version is not part of GoldED Light but it is part of the GoldED Pro(/NET) package; see Register GoldED . 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

hardware requirements

configuration

page header

printing

settings & units

page layout

page preview

files

ESC-sequences

character height

landscape

book

cover

single/doublesided

perforate

output device

shell interface

ARexx port HiSpeed

ARexx commands HiSpeed

macros

filter

sleep-icon

## 1.296 HARDWARE REQUIREMENTS

HiSpeed : HARDWARE REQUIREMENTS

Amiga, at least 512K, OS2.04, DeskJet/Laser. You'll need a softfont cartridge (a RAM expansion card) for your DeskJet if you want to use all functions of this program - otherwise you are limited to the printer's ROM fonts.

## 1.297 CONFIGURATION

HiSpeed : CONFIGURATION

HiSpeed has to be configured according to your printer - just choose your model from the list of supported printers. Set the RAM gadget to 'RAM' if your printer is equipped with a RAM expansion cartridge (DeskJet) or doesn't require additional RAM in order to use softfonts. The number of available typefaces depends on Printer/RAM selection - while a standard DeskJet without

RAM is limited to one typeface, a DeskJet 550C with RAM will offer 4 portrait fonts and 3 landscape fonts. If your printer is not supported by HiSpeed so far, you'll have to use the SetUp option to register your printer with HiSpeed:

HiSpeed may be adjusted to any PCL printer and its fonts. If your printer supports softfonts (DeskJet printers don't unless they are equipped with an additional RAM cartridge) you may use your own fonts with HiSpeed, too. Unfortunately setup isn't that easy. You should have read your printer's manual at least once. You have to be familiar with the use of PCL ESC codes. A bad setup will cause TROUBLE. Example session

Select the SETUP gadget. The setup requester will appear, giving you the opportunity to set up a printer and its fonts. On the left side available printers are listed. Select NEW (next to the list of printers) if your model isn't available so far: A new printer 'unnamed' will be added to the list. Use the string gadget below the list to change this name to whatever you like. Toggle the 'DeskJet tray' gadget to ON if your printer's paper tray looks like that of a DeskJet (HiSpeed will use this information to give you some hints if it asks you to reinsert ejected pages). Toggle the 'faces down' gadget to ON if the sheets are ejected faces down by your printer. Some printers (especially laser printers) are not able to use the first 5 to 7 mm of a page for output - turn OFFSET HEADER on in these cases. Finally you should select a file pattern used when selecting cover files (e.g. if you don't want HiSpeed to list color files). A pattern like '#?' would display all files while "~(#?.CYMK)" would suppress cover files made for the DeskJet 550C. The following extensions are supported:

```
CYM files (color, used by DJ500C/DJ550C)..... #?.CYM
CYMK files (color, used by DJ550C) ..... #?.CYMK
B&W files (PCL level 3) ..... no suffix
```

After having selected a printer, its fonts are displayed in the font listview on the right side. This list initially is empty if you have added a new printer. Use NEW (next to the font list) to add a new font (i.e. to register one of the printer's fonts with HiSpeed). New fonts are named 'unnamed'; use the string gadget below the list to edit the name. Having done this you have to enter a description of the font - HiSpeed needs to know the font's basic characteristics (e.g. its size). Only fixed width fonts are supported. Have a look at your printer's manual to determine all required values. Example:

If you own a DeskJet 550C, look for a manual section called 'Specifications'. There you'll find a list of available fonts supported by your printer. Among many other entries a 'LetterGothic' font with a font size of 6 points is described. According to the manual this fixed width font is available in portrait/landscape mode. Spacing is 24 cpi (characters per inch):

```
Typeface: LetterGothic
Point size: 6p
Spacing: 24 cpi (fixed width)
Orientation: portrait or landscape
```

Use these specifications to set up HiSpeed's font entry: Toggle the type gadget to 'any direction' since portrait as well as landscape mode are supported. The HMI gadget is used to set the font's spacing. Units are 1/1200", so you would enter 50 (1200/24 = 50). Specify the LPI (lines per inch) value for single linefeeds using the LPI gadget. Suggestion: Set the LPI value to 72/<point size> (e.g. 72/6 = 12). While the HMI value has to be

chosen as precisely as possible, the LPI may be set more roughly. If your font isn't a built-in one but a softfont file, enter the file name(s) using the NORM/ITAL gadgets (upright respectively italics version). Leave these gadgets empty if the font is a ROM font. Softfonts have to use the ECMA latin character set; the printer will refuse your fonts if they use a different character set (e.g. PC8). Finally we have to provide a PCL ESC sequence used by HiSpeed to activate the font. Have a look at you printer's documentation as far as PCL sequences (PCL = printer command language) are concerned. The sequence will have to set typeface, spacing and character height (point size). It mustn't set any other attribute. Example:

```
ESC(s6v24h6T
```

ESC introduces a PCL sequence. The last character of a PCL command has to be uppercase ('T') while all other characters (except ESC) have to be lowercase. The font height is set to 6 points by '(s6v'. '24h' sets spacing to 24 cpi. Finally '6T' selects a typeface called LetterGothic. Don't use spaces within PCL sequences. Pay attention to case.

## 1.298 PAGE HEADER

HiSpeed : PAGE HEADER

It is possible to change the text (or the style) of page headers using the setup requester, too. The header text is not printed directly. Instead, it is interpreted before output: after scanning it for keywords (like \DATE), found keywords are replaced by their 'current value' (e.g. the date). Put a text into quotation marks if you don't want to have it interpreted. HiSpeed does support the following keywords:

keyword	example	description
"<Text>"	"Hi!"	print quoted text
\DATE	"Date: " \DATE	insert date
\TIME	"Time: " \TIME	insert time
\NAME	"File: " \NAME	insert file name
\PATH	"Path: " \PATH	insert fully qualified name
\PAGE	"Page: " \PAGE	insert page number
\$<name>	\$USER	insert environment variable (*)

(\*): see AmigaDOS manual (SETENV command)

## 1.299 PRINTING

HiSpeed : PRINTING

The files ("jobs") to be printed are displayed as "job list" (bottom left). Scroll through that list using the arrow gadgets. Use 'Add' to add new jobs. A file requester will appear. You may add more than one file at once by holding down the shift key during selection. There are other methods to add

files to the job list, too: Just drag a file's icon over the HiSpeed window. Or set up a file via ARexx. Or let a file drop on HiSpeed's sleep icon (however, if you do so, it will be printed at once but not included into the job list).

Use job <menu/clear list> to clear the complete list respectively the 'del' gadget to remove a single file. Select "SORT A" if you want to have the list sorted alphabetically. Use "sort #" to sort by size.

Start printing by clicking at the print gadget. Jobs are processed from top to bottom. If none has been set up so far, you will be asked for a file. You may stop printing by closing the progress window (HiSpeed will ask for confirmation). Downloads mustn't be interrupted for this would confuse the printer.

It is possible to print single sheets, too, using the 'page' gadget. This option is useful to reprint some lost passages of a large job (e.g. if ink cartridge went empty during printing).

If you want a page header (file name, page number), activate "PAGE HEADER". See 'SetUp' on how to set up the header text (for example it is possible to have time/date strings or environment variables inserted). If you print more than one file, you might want an index: switch to header & index.

## 1.300 SETTINGS & UNITS

HiSpeed : SETTINGS & UNITS

Usually you won't have to change paper dimensions. Default settings are valid for output to A4. The left border should be sufficient for perforating. Paper dimension settings offered by the main window of HiSpeed only describe the area used for printing (e.g. if you want to restrict output to the upper half of a sheet, do it here). The actual physical paper size is set by a different requester (setup requester): usually to A4 in Europe respectively to US-Letter/US-Legal in the US.

Use "RESET ALL" to get the defaults. All settings are written to HiSpeed's configuration file by <project menu/save config>. Be careful as far as the dimensions setup is concerned - wrong values (especially insufficient borders) will confuse HiSpeed.

HiSpeed v5.0 or better does write its configuration file to "ENVARC:HISPEED" unless a different directory is specified. Former releases used to write to 's:' or directly to 'envarc:'. Those files should be removed.

Units are 'mm' with exception of the block size display - which uses <characters> to display text block dimensions. The block size depends on border/paper settings as well as font selection (font width).

## 1.301 PAGE LAYOUT

---

HiSpeed : PAGE LAYOUT

Layout definition (the number/arrangement of text blocks on a page) is of great importance for a pleasing result: The more block columns you order, the less characters will fit into a block line. If the block with is too small, lines have to be splitted.

Block width depends on the selected font, too. If you select Courier 6 (this is the only font offered by the standard DeskJet), no more than 75 characters will fit into a single line provided you use a layout of two columns. In generally 75 characters is not enough. You might use the Adjust-LF option in this case: It will remove LF (line feed) codes. However, don't use this option with formatted data (tables or source codes).

Edit contents of the dx/dy gadgets to set the distance between columns/rows. Enable the 'separate' option if you want to have a line drawn between two block rows. Blocks are linked from left to right and from top to bottom.

### 1.302 PAGE PREVIEW

HiSpeed : PAGE PREVIEW

HiSpeed offers two ways of quality/volume precalculation without having to waste a single page of paper: Use 'preview' to order a WYSIWYG preview. Or use 'scan' to get some statistics ( e.g. required number of sheets). While scanning is faster than a preview, the latter is more useful to get a general impression of the output. Use the cursor keys to flip through pages in preview mode. Use the help key to open the statistics requester.

The distortions value of the statistics display is related to text lines that were to long to fit into a single block line (i.e. lines that had to be splitted). Possible values are from 0% to 100%. This value should be as low as possible since splitted lines don't look that well. Decide for a smaller font, use less columns or decrease margins to avoid distortions.

### 1.303 FILES

HiSpeed : FILES

HiSpeed has been designed to process ASCII files with or without ANSI codes. Lines have to be terminated by LF codes. Additional CR codes (used by MS-DOS software) are recognized and removed. Turn the MS-DOS option ON if you want to print MS-DOS files (maybe received from a BBS system) - HiSpeed will use a different character set to cope with these files. Softfonts are not available in MS-DOS mode because they have been developed with the Amiga character set (ecma latin) in mind.

---

## 1.304 ESC-SEQUENCES

HiSpeed : ESC-SEQUENCES

Switch to HiSpeed mode or turn PERFORM ESC off if you want to have ESC-Sequences ignored (and thus speed up output). If you enable FFs (form feeds), every detected FF-code (12 dez) results in a new text block. You better have a prescan to estimate paper needs - some files (especially library manuals using the AutoDoc format) send FF codes after each paragraph.

HiSpeed mode does imply draft printing/normal stroke. Other modes use LQ printing, draft printing or bold printing. These settings stay valid until they are turned off by appropriate ANSI ESC sequences being part of the file(s) to print. They are not affected by ESC sequences if 'perform ESC' is off.

## 1.305 CHARACTER HEIGHT

HiSpeed : CHARACTER HEIGHT

Several fonts are available (number depends on your printer model), e.g. LinePrinter-5, LinePrinter-6, Courier-6 and LetterGothic-5. You should use LinePrinter-5 or L Gothic-6, if the average line width of your text files is greater than 80 characters. LinePrinter-6 looks better but block width might become too small (e.g. 80 characters if you choose a layout of two columns). Courier-6 is the worst choice: no more than about 75 characters will fit into a line if you decide for a layout of two columns.

## 1.306 LANDSCAPE

HiSpeed : LANDSCAPE

All DeskJet models apart from the original DeskJet do support landscape printing. The number of fonts available in landscape mode depends on your printer model.

## 1.307 BOOK

HiSpeed : BOOK

Output is sorted in a way to support manual creation if you enable book mode: simply fold printed sheets to get A5 manuals. HiSpeed relies on the user's settings: if you set the block width to 120 columns but the actual (average) line width of a text is 60 characters, output won't look good (columns are not centered). Simply run the AdjustBook macro after switching to book mode

---

to have the layout adjusted automatically. Since book mode usually implies landscape printing (not supported by the original DeskJet), this mode should only be used with a DeskJet+ or better.

### **1.308 COVER**

HiSpeed : COVER

Select 'cover' to order printing of A5 covers. Several templates are available. Quality of cover printing depends on the LQ/NLQ gadget. This function requires the XPK library. Coverfiles actually are PCL files (i.e. files usually sent to a printer). They have been crunched using xDrop of the XPK package. Recommended file extension for color files is \*.CYM (DeskJet 500C) or \*.CYMK (DeskJet 550C). Have a look at the templates directory if you are looking for the Pagestream documents used to create the original HS cover files.

### **1.309 SINGLE/DOUBLESIDED**

HiSpeed : SINGLE/DOUBLESIDED

Single sided as well as double sided printing is available. Don't get confused if HiSpeed appears to eject empty pages in double sided mode - those pages are required by the second pass of printing. Consider HiSpeed's advice if the program asks you to reinsert ejected pages in order to print the backsides. Handling depends on printing mode !

### **1.310 PERFORATE**

HiSpeed : PERFORATE

Enable 'perforate' if you want to have perforation considered during double-sided output, i.e. to make HiSpeed swap left/right borders if necessary to ensure a sufficient left margin.

### **1.311 OUTPUT DEVICE**

HiSpeed : OUTPUT DEVICE

Use the port gadget to select desired output device respectively output file. If output goes to a file, HiSpeed will create two files (even/odd pages) using file extension 1 respectively 2. File 1 has to be sent to your printer before file 2 later on since (only) this file will keep softfont data.

---

## 1.312 SHELL INTERFACE

HiSpeed : SHELL INTERFACE

HiSpeed can be used from a shell or be run by any program providing a shell startup. Supported arguments are according to the template scheme used by the AmigaDOS manual (the following options perform the same action as the gadgets of HiSpeed's GUI, so please have a look at the explanations above for details):

FILE/M, S=SLEEP/S, D=DOUBLE/K, RESET/S, H=HEADER/K, N=NUMBER/K, H=HISPEED/K, P=PREVIEW/S, C=SCAN/S, L=LANDSCAPE/K, B=BOOK/K, F=FEED/N, I=INDEX/S, LQ/S, EDV/S, BOLD/S, ADJUSTLF/K, SORTNAME/S, SORTSIZE/S, ASL/S, SCREEN/K, PERFORATE/K, CMDFILE/K, MSDOS/K, UNI/K, ANSI/K, PARSEFF/K, FILTER/K, PAR:/S, SER:/S

## 1.313 AREXX PORT HISPEED

HiSpeed : AREXX PORT HISPEED

HiSpeed offers an ARexx port. Using ARexx it is possible to add headers to files before printing them, to format a text, to generate an appendix or to move HiSpeed to a different (public) screen. Usually you can operate the program from WB and ARexx ( AREXX COMMANDS HISPEED ) simultaneously.

## 1.314 ARexx commands HiSpeed

HiSpeed : ARexx commands HiSpeed

Group 1 - set variables (SET); Syntax: SET <variable> <value>

```

SET LAYOUTX <columns>..... set columns
SET LAYOUTY <rows>..... set rows
SET LEFT <mm>..... left border
SET RIGHT <mm>..... right border
SET TOP <mm>..... top border
SET BOTTOM <mm>..... bottom border
SET DX <mm>..... space between columns
SET DY <mm>..... space between rows
SET COPIES <number>..... copies (max. 99)
SET FEED <0...12>..... linefeed
SET PORT <'PAR:'|'SER:'|FILE>..... output port
SET OUTFILE '<full path>'..... output file
SET QUALITY <HISPEED|LQ|EDV|BOLD|BOLDEDV>... style
SET FILE '<full path>' ..... add file to list
SET PERFORMESC <ON|OFF> ..... perform ESC codes ?
SET PERFORMFF <ON|OFF> ..... perform FF codes ?
SET NUMBER <ON|OFF> ..... number lines ?
SET HEADER <OFF|NOHEADER|ON|INDEX> ..... header type
SET SIZE <Fontname> ..... font
SET DOUBLE <ON|OFF> ..... double sided ?

```

SET SINGLE <ON|OFF> ..... ditto  
 SET SEPARATE <ON|OFF> ..... draw lines ?  
 SET ADJUSTLF <ON|OFF> ..... ignore LF ?  
 SET KEEPLF <ON|OFF> ..... perform LF ?  
 SET REQUESTER <ON|OFF> ..... no requesters ? (1)  
 SET ASK '<Text>' ..... show YES/NO requester  
 SET WARN '<Text>' ..... show I SEE requester  
 SET BOOK <ON|OFF> ..... toggle book mode  
 SET COVER <coverfile> ..... print a cover  
 SET KEYLIST <file> ..... keyword file (\*)  
 SET KEYCODE <character|OFF> ..... set keyword qualifier  
 SET PAPERX ..... set paper width  
 SET PAPERY ..... set paper height  
 SET PERFORATE <ON|OFF> ..... prepare perforation ?  
 SET ORIENTATION <PORTRAIT|LANDSCAPE> ..... paper orientation  
 SET MSDOS <ON|OFF> ..... toggle MS-DOS mode  
 SET UNIDIR <ON|OFF> ..... uni/bidirectional  
 SET SCREEN <PubScreenName> ..... select a screen

#### Group 2 - subroutines

QUIT ..... quit HiSpeed  
 SLEEP ..... iconify  
 OPENWIN ..... open main window  
 SORTNAME ..... sort jobs by name  
 SORTSIZE ..... sort jobs by size  
 NEW ..... clear job list  
 CLR ..... ditto  
 REQUESTFILE ..... ask for files  
 PRINT ..... print jobs  
 SCAN ..... scan jobs  
 VSCAN ..... ditto (visualize)  
 RESET ..... get default settings  
 INFO ..... show scan results  
 SAVECONFIG ..... save settings

#### Group 3 - read variables (QUERY)

Syntax: QUERY <variable> <qualifier> (result is placed in RESULT)

QUERY LAYOUTX ..... layout (columns)  
 QUERY LAYOUTY ..... layout (rows)  
 QUERY LEFT ..... left border  
 QUERY RIGHT ..... right border  
 QUERY TOP ..... top border  
 QUERY BOTTOM ..... bottom border  
 QUERY DX ..... space between columns  
 QUERY DY ..... space between rows  
 QUERY COPIES ..... number of copies  
 QUERY FEED ..... linefeed  
 QUERY PORT ..... port  
 QUERY OUTFILE ..... output file (port=FILE)  
 QUERY QUALITY ..... style of printing  
 QUERY PERFORMESC ..... perform ESC codes ?  
 QUERY PERFORMFF ..... perform FF codes ?  
 QUERY NUMBER ..... number lines ?  
 QUERY HEADER ..... header type

```

QUERY SIZE ..... font
QUERY DOUBLE ..... printing mode
QUERY SINGLE ..... ditto
QUERY SEPARATE ..... draw lines ?
QUERY ADJUSTLF ..... adjust-LF setting
QUERY KEEPLF ..... ditto
QUERY BLOCKX ..... characters/line
QUERY BLOCKY ..... lines/block
QUERY PAGES ..... required pages (*)
QUERY SHEETS ..... required sheets (*)
QUERY DISTORTIONS ..... get distortions (*)
QUERY BYTES ..... output size (bytes)
QUERY JOBS ..... # of files to print
QUERY VERSION ..... get version string
QUERY JOBNAME <slot> ..... get a job's name
QUERY STATE ..... read HiSpeed's state
QUERY JOBPAGE <slot> ..... get job's page (*)
QUERY BOOK ..... book mode set ?
QUERY MAXLINE ..... longest line (*)
QUERY KEYCODE ..... keyword qualifier
QUERY PERFORATE ..... border handling
QUERY ORIENTATION ..... paper orientation
QUERY MSDOS ..... MS-DOS mode ?
QUERY UNIDIR ..... uni/bidirectional ?

```

(1) supported only within ARexx scripts

(\*) these variables become valid after a SCAN (or VSCAN)

## 1.315 MACROS

HiSpeed : MACROS

ARexx macros found within the HISPEED/AREXX directory during startup are gathered as MACRO menu. Please read the 'ARexxEDoc' manual for more details about ARexx.

## 1.316 FILTER

HiSpeed : FILTER

Toggle 'FILTER' ON if you want the '@' character to be regarded as keyword qualifier. It will be removed before output. Have a look at the MakeAppendix macro.

## 1.317 SLEEP-ICON

HiSpeed : SLEEP-ICON

Use the 'Zzz' gadget to iconify HiSpeed. You may drag files over the icon to have them printed - it's an AppIcon. Doubleclick the icon to wake HiSpeed up again. Since the Amiga OS does not support reading AppIcon coordinates, HiSpeed won't be able to remember the icon position after you have moved it to another location. Nevertheless it is possible to snapshot the preferred position: Open the 'hispeed:config' 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. HiSpeed will read the new position the next time it is evoked.

## 1.318 RECOVER

Tools : RECOVER

A program to recover text buffers after your system has crashed. Recover will scan all available memory locations (the free memory pool as well as memory allocated by other task), looking for lost buffers. Since freed memory is not protected by the OS, lost buffers may become trashed. Recover will still try to restore trashed buffers but they may contain garbage lines. You'll have to verify that restored buffers are intact before you copy them over your old files. Recover will try to figure out how many lines are corrupt to give you a rough idea of whether additional work is required. Be warned: these figures are not exact since it's hard to determine whether a line is corrupt or not. To increase your chances, run this program as soon as possible. Interrupt your startup sequence (type CTRL-D) and run recover IMMEDIATELY. Recover disables multitasking while scanning memory to prevent other tasks from allocating memory - this means that your mouse pointer freezes temporarily. Recover should be run from a shell window. The following arguments are supported:

RECOVER DRIVE/K,ALL/S,TEST/S,MAXLEN/N,MAXLINES/N

DRIVE

Drive where to write restored files (e.g. df0:). Don't use a ram disk (might overwrite the text to recover) or a hard disk (program might fail while writing since it has to do some non-system-friendly stuff).

ALL

Recover file even if it appears to be unchanged (Recover defaults to ignoring a text if it has never been modified).

TEST

Just scan memory for text buffers to recover but don't write to a disk. Recover tries to figure out how many lines can be restored respectively are corrupt. A line is considered corrupt if it contains ASCII-Codes below 32 or from 128 to 160. Nevertheless, even corrupt lines are restored since parts of them may still be usable.

MAXLEN

---

Maximum line length to be considered valid. If Recover happens to encounter a longer line while scanning memory it will refuse to restore the line (might be corrupt). Defaults to 255.

#### MAXLINES

Maximum number of lines (per text) to restore. Recover will stop recovery as soon as a text exceeds this limit (text might be corrupt). Defaults to 10000 lines.

## 1.319 REGISTER GOLDED

### REGISTER GOLDED

A GoldED keyfile or the GoldED software and a keyfile are mailed to you on disk(s) or send to your e-mail address if you register. The keyfile removes the 1000-lines restriction from the trial version, turning it into a registered version. Please use the Order program (available in the GoldED drawer) to print your order. Addresses for support, orders and bug reports are listed on the following pages ( address ).

Delivery usually can be expected in four to eight weeks, under special circumstances delivery may take longer. A keyfile is e-mailed to you in advance (if requested) to allow you working with GoldED while waiting for your disk(s). Keyfile requests should include your full address and your PGP key. Requests without a PGP key are not processed. PGP is a freeware encryption utility. Please read the PGP documentation if you haven't used PGP before.

Eurochecks (made out in DM) and cash are accepted as payment for GoldED. Postal money cash orders are accepted, too. Other checks (American Express Checks, Visa Checks, Traveller Checks, Postal money Order Checks) are not accepted. Payment via a bank account is available for GoldED NET only (Dietmar Eilert, account 5129 92-505, Postbank Köln, BLZ 370 100 50). All offers valid until end of 3/1997.

Special discounts are available if you order more than one copy (don't forget to provide a user name/address for each registration):

2 registrations ..... minus 20% each  
3 or more registrations ..... minus 30% each

GoldED Light (29.90 DM Germany, 35 DM Europe, 25\$ international)

-----  
Mailing consists of one disk. Includes a keyfile for GoldED (only). The latest versions of the GoldED package as well as standard disk-based documentation is shipped. Please state whether you own a 1.7 MB HD disk drive.

GoldED Pro (39.90 DM Deutschland, 45 DM Europa, 35\$ international)

-----  
Mailing consists of one or two disks and a printed manual of about 70 pages.

---

Please state whether you own a 1.7 MB HD disk drive. Includes a single user licence for the latest versions of GoldED & HiSpeed. Please choose the manual translation you would like to receive (English and German are available).

GoldED NET (25 DM international)

-----

The net distribution: you'll receive a PGP-encrypted keyfile for GoldED and HiSpeed at your e-mail address. You can use your currently installed unregistered GoldED/HiSpeed (4.0.0 or better) without restrictions after installing the keyfile. Send your order to dietmar@tomate.tng.oche.de. Include your full name and address and your PGP key (pgp -kxa). Orders without a PGP key are not processed.

## 1.320 UPDATES

### UPDATES

Registered GoldED users have received a keyfile for GoldED, allowing them to use future updates without restrictions. However, purchase of GoldED doesn't include unlimited access to free updates: the author may cease to provide free updates or updates at all without prior notice. Major updates may or may not require payment of a reasonable update fee. Users of GoldED OEM versions will have to return the GoldED registration card first in order to receive support or access to updates (please read the documentation of your software).

GoldED updates usually are available at AmiNet sites and at the GoldED WWW support site ( WWW support ) free of charge. Updates on disk can be ordered ( address ) for a fee of 15 DM, disk and postage (Europe) included. Prices for delivery into other parts of the world available on request.

## 1.321 Upgrading to GoldED 4

### Upgrading to GoldED 4

GoldED 4 doesn't accept keyfiles made for former GoldED distributions: a new keyfile is required to use GoldED 4. Keyfile updates are free if you have received your GoldED registration between 1/96 and 8/96: the new keyfile is e-mailed to your e-mail address after we have received your PGP key. Delivery on disks is available, too: return your original disk and a self addressed envelope, postage paid (postal reply coupon, available at your post office).

Updates of GoldED registrations received before 1/96 are available for an update fee of 15 DM, disk and European postage included. Prices for delivery into other parts of the world on request. Eurochecks, postal money cash orders and cash are accepted as payment. Other checks are not accepted. You may choose between delivery to your e-mail address (new keyfile only, PGP key required) or delivery on disk. Delivery on disk includes the latest GoldED version, e-mail orders are processed faster.

---

## 1.322 CREDITS

### CREDITS

\* DICE \* Reqtools \* XPK \* ARexxBox \* GadToolsBox \*

This program has been developed using Matt Dillon's Dice. Most of the requester design has been done using Jan van den Baard's GadToolsBox. Thanks to Nico François for his Reqtools library and to the developers of the XPK (de)compression system. The ARexx routines of GoldED have been inspired by source code created by Michael Balzer's ARexxBox (though less sophisticated). Thanks to Stefan Zeiger for Boopsi example source code. And thanks to Joerg Gutzke, Dario Fava & Thomas Lechner, sysops of the Mowgli BBS, Mailway BBS and Tomate BBS (sources of most of the tools mentioned above). GUIMake has been developed by Rico Krasowski. Finally, I would like to thank these people for their invaluable suggestions, translations, ideas & support: Giovanni Addabbo, Henric Andersson, Jochen Becher, Markus Aretz, Cristian Castellari, Martin Fay, David 'Edi' Göhler, Georges Goncalves, Christian Gottschling, Serge Hammouche, Andreas Harrenberg, Mick Hohmann, Henning Hucke, René Laederach, Lieven Lema, Lars Renström, Maarten Ter Mors, Rodolphe Sanderson (French translations) and Stefan Schor. Further acknowledgements go to the following people who created many useful GoldED ARexx scripts: Eric Burghard, Oliver Clouth, Leu Simon Gris, Francois Helsen, Tattoo Mabonzo, Krzysztof P. Jasiutowicz, Fin Schuppenhauer and Markus Zahn.

## 1.323 ADDRESS

### ADDRESS

Bugreports, comments and suggestions (e-mail, phone) are welcome. However, personal support is available for registered users only: please name the registration you own. Users of GoldED OEM versions will have to return the GoldED registration card first in order to receive support or access to updates (please read the documentation of your software). Users of a GoldED trial version can find some GoldED-related information at the WWW site (information about new versions, prices).

### WWW SUPPORT

Visit our Internet WWW support site. GoldED related news, updates, tools:

<http://www.clearlight.com/~dietmar>

### SUPPORT FOR UNREGISTERED USERS

A file server has been set up to answer standard questions. The file server responds to e-mails. Send an e-mail to the file server address with the subject set to "send file info" to receive further information. E-mail address of the file server:

[dietmar@clearlight.com](mailto:dietmar@clearlight.com)

### BUG REPORTS, SUGGESTIONS

Suggestions are welcome but usually are not answered (due to lack of time). Questions related to if or when special features are implemented can not be answered - new features and new versions are announced at the WWW support instead. Bugreports are answered if the bugs can be reproduced or if the quality of the report (please include enforcer report, mungwall report, snoopdos log) suggests that there actually is a bug to be fixed.

E-mail address: Dietmar.Eilert@post.rwth-aachen.de

#### ORDERS, SUPPORT FOR REGISTERED USERS

Technical questions and orders should be send to the address below. Bug reports, suggestions or non-technical questions should not be send to this address (and are ignored if send to this address).

Dietmar Eilert  
Mies-v-d-Rohe-Str. 31  
D-52074 Aachen  
Germany  
E-Mail: dietmar@tomate.tng.oche.de

Phone: 0241/81665  
Fax: 0241-81665-22 (+49-241-81665)

### 1.324 GoldED

ABOUT GOLDED	FILETYPES	PROJECT/CLEAR WINDOW
ADDRESS	FILTER	PROJECT/CURRENT DIR
APC	FIND	PROJECT/CURRENT DIR...
API	FIRST	PROJECT/INFORMATION
APPWINDOWS	FIX	PROJECT/INSERT
AREXX COMMANDS HISPEED	FOLD	PROJECT/NAME
AREXX PORT	FOLDING	PROJECT/OPEN
AREXX PORT HISPEED	FONT	PROJECT/OPEN FAST
AREXXBOX	FORMAT	PROJECT/OPEN NEW
ARGUMENTS	FORMATTER	PROJECT/OPEN ORIGINAL
AUTOARRANGE	FREEZE	PROJECT/PRINT
AUTOBACKUP	FULL SCREEN	PROJECT/PROTECTION
AUTOCASE	FUNC	PROJECT/QUIT ALL & EXIT
AUTOCOMPLETION	FUP	PROJECT/QUIT BUFFER
AUTOFOLD	GADTOOLSBOX	PROJECT/SAVE
AUTOINDENTION	GETTING STARTED	PROJECT/SAVE AND CLOSE
BACK	GLOBAL SEARCH	PROJECT/SAVE AS
BEEP	GOTO	PROJECT/SAVE AS XPK
BIND	GREP	PUSH
BITS	GUI	QUERY
BLOCK	HARDWARE REQUIREMENTS	QUICKFUNC
BLOCK MENU	HELP	QUICKREFERENCE
BLOCK/APPEND TEXT	HELP KEY	QUICKSTARTER
BLOCK/BLOCK PRINT	HIDE	QUIT
BLOCK/BLOCK SAVE	HISPEED	RECOVER
BLOCK/BLOCK SORT	HOTKEY	REDO
BLOCK/BLOCK TO LOWERCASE	HOW TO FOLD LINES	REFRESH
	HUNTER	REGISTER GOLDED

BLOCK/BLOCK TO UPPERCASE	IF	REMAP
BLOCK/CLIPBOARD COPY	INDENT	REMOVAL OF COLUMNS
BLOCK/CLIPBOARD CUT	INDEX	REPLACE
BLOCK/CLIPBOARD PASTE	INFO	REQLIST
BLOCK/COLUMN	INPUT EVENTS	REQTOOLS
BLOCK/COLUMN DELETE	INSERT	REQUEST
BLOCK/COLUMN INSERT	INSERTION OF COLUMNS	REQUIRED SYSTEM
BLOCK/COLUMN TEXT	INTELLISENSE	RETURN KEY
BLOCK/COPY	INTERNAL COMMANDS	RIGHT
BLOCK/DELETE	INTRODUCTION	RUN
BLOCK/MARK	KEY	RX
BLOCK/MARK HIDE	KEYBOARD	SAVE
BLOCK/MARK LINE	LANDSCAPE	SCREEN
BLOCK/MARK PARAGRAPH	LANGUAGE	SCROLL BORDERS
BLOCK/MOVE	LAYOUT	SEARCH MENU
BLOCK/PASTE COLUMN	LAYOUT MENU	SEARCH/ASCII CODE ?
BOOK	LAYOUT/AUTOCASE	SEARCH/ASCII CODE INSERT
BRACKET	LAYOUT/BLOCK CENTERED	SEARCH/ASCII TABLE
BREAKPT	LAYOUT/BLOCK JUSTIFIED	SEARCH/CHARACTER SET
CENTER WINDOWS	LAYOUT/BLOCK LEFT	SEARCH/CHECK
CHARACTER HEIGHT	LAYOUT/BLOCK RIGHT	SEARCH/COMPLETE
CHARACTER SET REMAP	LAYOUT/INDENT	SEARCH/COUNT PATTERN
CHUNKY PIXEL	LAYOUT/INDENT BLOCK	SEARCH/FIND
CLIP	LAYOUT/INDENT LINE	SEARCH/FIND NEXT
CLIPBOARD	LAYOUT/INSERT MODE	SEARCH/FIND PREVIOUS
CMD	LAYOUT/LINE: LEFT	SEARCH/FUNCTION FIND
CODE	LAYOUT/LINE: RIGHT	SEARCH/FUNCTION LIST
COLON	LAYOUT/NUMPAD = MOVEMENT	SEARCH/MATCHING BRACKET
COMMAND LIST	LAYOUT/PARAGRAPH CENTERED	SEARCH/REFERENCE
CONFIGURABLE GADGETS	LAYOUT/PARAGRAPH JUSTIFIE	SEARCH/REFERENCE...
CONFIGURATION	LAYOUT/PARAGRAPH LEFT	SEARCH/REPLACE
CONSOLE	LAYOUT/PARAGRAPH RIGHT	SEARCH/REPLACE HISTORY
COVER	LAYOUT/SET RIGHT MARGIN	SEARCH/REPLACE NEXT
CR	LAYOUT/SOLID TABS	SELECT A HOST
CREDITS	LAYOUT/TEMPLATES	SEQUENCES
CURSOR KEYS	LAYOUT/USE CURRENT MARGIN	SESSION
DEBUG	LAYOUT/WORD WRAP	SET
DEBUGGER	LEFT	SETTINGS & UNITS
DEL	LICENCE	SHELL INTERFACE
DEL KEY	LINES	SHIFT
DELETE	LOCK	SHIFTING
DICE	LOCK A WINDOW	SINGLE/DOUBLESIDED
DIR	MACRO	SLEEP-ICON
DISPLAY MODE	MACRO MENU	SMARTCR
DJUMP	MACROS	SMARTINDENTION
DO YOUR JOB	MACROS/EXECUTE COMMAND	SPACE KEY
DOWN	MACROS/EXECUTE LINE	SPOOLER
DPAGE	MACROS/MACRO EDIT	STOP
DRAG & DROP	MACROS/MACRO EXECUTE	SUFFIX
DYNAMIC TABS	MACROS/MACRO START	SYNTAX
ELSE	MACROS/MACROS	SYNTAX HIGHLIGHTING
ENDIF	MACROS/SEQUENCE LOAD	TAB
ENDWORD	MACROS/SEQUENCE PLAY	TAB KEY
EOL WRAP	MACROS/SEQUENCE PLAY...	TABS
ERROR BEEP	MACROS/SEQUENCE RECORD	TASK
ESC KEY	MACROS/SEQUENCE SAVE	TEMPLATES
ESC-SEQUENCES	MACROS/SESSION LOAD	TEXT

---

EVENT DEFINITION	MACROS/SESSION SAVE	TMPLATE
EXALL	MAGIC CODES	TOOLBAR
EXCLUDE CURSOR	MAIN	TOOLS
EXTRACT	MARGINS	TYPE
EXTRAS MENU	MARK	UJUMP
EXTRAS/AMIGADOS SHELL	MAXDOWN	UNDO
EXTRAS/FILES	MAXUP	UNDO & REDO
EXTRAS/HISPEED	MENU	UNLOCK
EXTRAS/INSERT TEXT	MENU CONFIGURATION	UNLOCK GUI
EXTRAS/LINE DOUBLE	MENU TREE OF BLOCK MENU	UP
EXTRAS/LINE INSERT	MENU TREE OF EXTRAS MENU	UPAGE
EXTRAS/LINE REMOVE	MENU TREE OF LAYOUT MENU	UPDATES
EXTRAS/LINE SWAP	MENU TREE OF MACRO MENU	UPGRADING TO GOLDED 4
EXTRAS/OPTIONS GLOBAL	MENU TREE OF PROJECT MENU	USE
EXTRAS/OPTIONS GLOBAL/FIL	MENU TREE OF SEARCH MENU	VARIABLES
EXTRAS/OPTIONS GLOBAL/GUI	MENU TREE OF VIEW MENU	VIEW
EXTRAS/OPTIONS GLOBAL/MIS	MENUHELP	VIEW MENU
EXTRAS/OPTIONS GLOBAL/VAR	MENUS	VIEW/ARRANGE HORIZONTAL
EXTRAS/OPTIONS LOCAL	MISC	VIEW/ARRANGE VERTICAL
EXTRAS/OPTIONS LOCAL/API	MORE	VIEW/DEBUGGER
EXTRAS/OPTIONS LOCAL/DISP	MOUSE	VIEW/FOLD
EXTRAS/OPTIONS LOCAL/GADG	MOUSE CONFIGURATION	VIEW/GO TO LINE
EXTRAS/OPTIONS LOCAL/GUI	MULTISELECTION	VIEW/GO TO MODIFICATION
EXTRAS/OPTIONS LOCAL/HUNT	NAME	VIEW/GO TO TOP-BOTTOM
EXTRAS/OPTIONS LOCAL/INDE	NEW	VIEW/HIDDEN BUFFERS
EXTRAS/OPTIONS LOCAL/KEYB	NEXT	VIEW/NEW WINDOW
EXTRAS/OPTIONS LOCAL/LAYO	NOTIFY	VIEW/POSITION RECALL
EXTRAS/OPTIONS LOCAL/MENU	OPEN	VIEW/POSITION STORE
EXTRAS/OPTIONS LOCAL/MISC	OPTIONS GLOBAL	VIEW/PREVIEW
EXTRAS/OPTIONS LOCAL/MOUS	OPTIONS LOCAL	VIEW/REMEMBER SIZE
EXTRAS/OPTIONS LOCAL/PRIN	OUTPUT DEVICE	VIEW/SYNTAX HIGHLIGHTING
EXTRAS/OPTIONS LOCAL/PROJ	PAGE HEADER	VIEW/TOOLBAR
EXTRAS/OPTIONS LOCAL/REFE	PAGE LAYOUT	VIEW/WINDOW
EXTRAS/OPTIONS LOCAL/SYNT	PAGE PREVIEW	VIEW/WINDOW ARRANGE
EXTRAS/OPTIONS LOCAL/TABS	PARAGRAPH VS. BLOCK	VIEW/WINDOW CENTER
EXTRAS/OPTIONS LOCAL/TEMP	PARENTHESIS CHECK	VIEW/WINDOW MAXIMIZE
EXTRAS/OPTIONS LOCAL/WORD	PATH	VIEW/WINDOW NEXT
EXTRAS/PROJECT FILTER	PC	VIEW/WINDOW PREVIOUS
EXTRAS/REDO	PERFORATE	VIEW/WINDOW ZIP
EXTRAS/SAVE PREFERENCES	PHRASE	VLEFT
EXTRAS/SEARCH FILE	PING	VRIGHT
EXTRAS/SEARCH FILE...	PONG	WEIGHT
EXTRAS/UNDO	POP	WHITE SPACE
F-KEYS	PREFS	WILDCARDS
FAST SCROLLING	PREV	WINDOW
FASTLOAD	PREVEND	WORD
FDOWN	PREVIEW	WORDWRAP
FEATURES	PRINT	WWW SUPPORT
FILE	PRINTER	XPK
FILE HUNTER	PRINTING	XPK SUPPORT
FILE LIST	PROJECT	XREF
FILES	PROJECT MENU	
FILETYPES	PROJECT/APPEND	