golded

dietmar eilert

COLLABORATORS

	TITLE : golded	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	dietmar eilert	April 14, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

golded

1.1 main

1.2 Licence

Licence

1a. END-USER LICENSE AGREEMENT

IMPORTANT - READ CAREFULLY: This end-user license agreement ("LICENSE") is a legal agreement between you (either an individual or a single entity) and Dietmar Eilert ("GOLDED SUPPORT") for the software product identified above, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT" or "SOFTWARE"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this LICENSE. If you do not agree to the terms of this LICENSE, you may not use the SOFTWARE PRODUCT.

SOFTWARE PRODUCT LICENSE

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. This LICENSE grants you the following rights:

- Software. You may install and use the SOFTWARE PRODUCT on any computer located on your premises for which you own an AmigaOS licence.

- Storage/Network Use. You may also store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT on any computer for which you own an AmigaOS licence over an internal network.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

- Limitations on Reverse Engineering, Decompilation, and Disassembly. You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

- Separation of Components. The SOFTWARE PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer.

- Rental. You may not rent or lease the SOFTWARE PRODUCT.

- Software Transfer. You may permanently transfer all of your rights under this LICENSE, provided you retain no copies, you transfer all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades, this LICENSE, and, if applicable, the Certificate of Authenticity), and the recipient agrees to the terms of this LICENSE.

- Termination. Without prejudice to any other rights, GOLDED SUPPORT may terminate this LICENSE if you fail to comply with the terms and conditions of this LICENSE. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

3. COPYRIGHT

- All title and copyrights in and to the SOFTWARE PRODUCT (including but not limited to any images, text, and "applets" incorporated into the SOFTWARE PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PRODUCT are owned by GOLDED SUPPORT or its suppliers. The SOFTWARE PRODUCT is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE PRODUCT like any other copyrighted material. You may not copy the printed materials accompanying the SOFTWARE PRODUCT.

MISCELLANEOUS

- If you acquired this product in Germany, this LICENSE is governed by the laws of Germany. If this product was acquired outside Germany, then local law may apply.

- NO WARRANTIES. The SOFTWARE PRODUCT and any related documentation is provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties or merchantability, fitness for a particular purpose, or noninfringement. The entire risk arising out of use or performance of the SOFTWARE PRODUCT remains with you.

- NO LIABILITY FOR DAMAGES. In no event shall GOLDED SUPPORT or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use this product, even if GOLDED SUPPORT has been advised of the possibility of such damages. Because some states/jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Aachen, October 1998, Dietmar Eilert

1.3 Introduction

Introduction

This editor is targeted at the serious software developer: it provides a fast, reliable and comfortable development environment. Installation and usage have been designed to be simple and clear, the user interface is functional and unobtrusive. All features of this program have been optimized for speed and comfort, often at the expense of above-average hardware requirements. While GoldED's hardware requirements are easily met by every developer machine, you will not be able to fully enjoy it on a low-end setup: GoldED has been designed to put a developer machine to good use, not to comply with the performance of entry-level computers.

A few words on the internal design of this software will help you to better understand the rest of this manual: the core of GoldED is event based. This means that the editor waits for events like a keystroke, a menu selection, etc. and then calls a dispatcher to perform the "action" associated with the event. This "action" is not hard-coded into the program: you can freely configure what happens when a menu is selected, a key is pressed, etc. For example, you can assign the "open file" function to the <A> key by using GoldED's keyboard configuration dialog. Or assign the text "Don't panic" to the A key. Or assign a macro script to it. Or an AmigaDOS command. Or just leave it as it is. The event-based design is extremely flexible because almost everything in GoldED can be configured to meet your demands.

1.4 Requirements

Requirements

Minimum requirements are AmigaOS 3, a 68020 CPU and 2 MB free RAM (when starting the program) but this configuration does not ensure acceptable performance. The suggested minimum system for average performance is a 68030 CPU and 4 MB free RAM. Total memory requirements including memory consumed by a compiler and the OS are between 8 MB RAM and 16 MB RAM.

Software Requirements

GoldED uses the XPK compression libraries if installed on your computer. The XPK libraries can be found on Aminet CDs. Installation is not required for normal usage of GoldED.

GoldED supports graphics cards

Adding a graphics card to your system improves performance drastically. For example, the PicassoIV board scrolls text with 256 colors (on a 800x600 screen) ten times faster than the same Amiga 3000 without a graphics card (720x480 pixel and 16 colors).

1.5 Getting Started

Getting Started

You could start this editor as you start most other programs: doubleclick on the icon or type the name in a shell window ("golded:golded" or "run golded:golded"). However, these methods are not recommended to start GoldED: the main disadvantage is that a new instance of the program is started every time you run GoldED, consuming memory and other resources. We recommend that you use the QuickStarter ED from the GoldED drawer instead:

QuickStarter

The quickstarter ED is a small (4 KB) GoldED frontend. You can use it as if it were the real editor. For example, you could enter "ed letter" in a shell window to edit the file "letter". The advantage of the starter is its ability to pass a new job to a running instance of GoldED. A new instance of GoldED is started automatically if necessary. Another advantage is that it supports both, detaching and non-detaching startup. The starter, unlike GoldED itself, may be made resident. The source code of ED is available in the GoldED drawer (golded:developer/examples/quickstarter).

Arguments

The starter and GoldED support various startup arguments (besides a list of files to be edited):

HIDE/S

Use the HIDE option (but do not specify file names) to start the editor as background process: the editor won’t open a window but will stay in the background and wait for you to press the Hotkey . Example:

```
ED HIDE
```

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

This option overrides automatic filetype detection: the example above makes GoldED use the configuration created for files of the type "TEXT" no matter what the real filetype of the specified file "letter" is (it could be an e-mail). Technically speaking, the argument after FILETYPE ("TEXT") is used as virtual file name (instead of the real file name which would be "letter") during filetype dection. See Filetypes for more information on filetypes.

STICKY/S (supported by the quickstarter only)

The quickstarter supports synchronous operation (non-detaching startup) and asynchronous operation (detaching startup) depending on whether or not the STICKY option is used. A call to ED returns immediately if STICKY is not specified. Specify STICKY if you want the quickstarter to return after the window opened by ED is closed: ED returns when the user quits out of the file. This option is designed to be used to synchronize GoldED with a program running ED to edit a file. Typical usage: when using ED with an e-mail program, the STICKY option should be used so that the editor waits until you have completed editing the e-mail before passing control back to the e-mail program. Example:

```
ED s:user-startup STICKY
```

AREXX/K

This option sets the name of GoldED's Rexx port. The AREXX/K option can only be used if GoldED is not yet running (you can't change the name of an open Rexx port with this command). The default name for GoldED's Rexx port is "GOLDED.1" ("GOLDED.<n>" for the nth instance of GoldED). Example:

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

SESSION/K

The SESSION comand restores the environment of a previous session (as saved by the
Macros/Save Session
menu). Example:

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

Drag & Drop starter

The drag & drop starter from the GoldED drawer creates a drag & drop icon on the Workbench. You can start the editor by clicking on the icon. You can drag text files over the icon to load them into the editor. We recommend to place the drag & drop starter in the wbstartup drawer of the Workbench so that it is started automatically when booting your computer. The icon has a small menu which permits saving the icon position. You'll see the menu if you activate the icon (click on the yellow arrow which is a hidden title bar). Use the right mouse button to select a function from the menu.

1.6 User Interface

User Interface

The editor's user interface supports most features of the AmigaOS you are used to. Additionally, it provides features taken from the Windows world.

MenuHelp

The editor's help facility is based on the AmigaGuide library. Doubleclick the "Manual" icon to view the manual with the Multiview program. GoldED provides online help for menus: press the <HELP> key while selecting a menu to get explanations related to the item you have selected.

Language

The locale library is supported: button labels and window titles should appear in the language you have chosen for your Workbench provided that the appropriate translations are shipped with GoldED. Currently we include German and English translations with GoldED. Other languages might become available in the near future. Locale settings do not affect menus and other configurable elements of GoldED: The language for these elements is choosen during the first installation of GoldED (the preferred language is saved to the file golded:etc/env/language).

Shortcuts

Most buttons have a keyboard shortcut. Underlined characters in button labels show the shortcut. Shortcuts allow you to move a slider or to activate buttons without using your mouse. Hold the SHIFT key down to toggle the direction of changes (e.g. to move a slider button one step to the left instead of one step to the right). Use the AMIGA key with the shortcut to activate another button while the cursor is in a string field (the keystroke would go to the string field if you don't use the AMIGA key simultaneously).

Text fields

Text fields are different from classic Amiga text fields. They support cut & paste. You can mark text with your mouse, cut and copy text (STRG-X or STRG-C) and insert text from the clipboard (STRG-V). Text input automatically replaces marked text in a string field. Move the cursor do hide the marker if you do not want this to happen. AlT-DEL deletes the word under the cursor, SHIFT-DEL deletes from the cursor position to the end of the text and SHIFT-BACKSPACE from the cursor position to the start of the text. SHIFT plus one of the cursor keys moves the cursor to the next or previous word. Note: on some keyboards, the STRG key is labeled "CTRL" and vice versa.

Lists and Trees

Most entries in lists and trees can be renamed. Select an entry with your mouse and click again on its name. A text field should appear where you have clicked if the entry can be renamed. There has to be a small pause between the two clicks so that the editor doesn't interpret it as a doubleclick.

Drag & Drop

All lists and trees support drag and drop. Click on an entry and hold the mouse key down for a moment. The color of the marker changes and you can drag the selected entry to a new position. However, using the drag & drop mechanism should be avoided: all lists have arrow buttons which should be used to move the selected entry. The problem with drag & drop is that it permits you do drag anything anywhere, possible corrupting the configuration. The arrow buttons do permit valid operations only.

Floating requesters

Some dialog boxes (the find requester, the goto requester, etc.) can be left open permanently: click on the "x" gadget in the window border to leave these requesters open.

1.7 Mouse buttons

Mouse buttons

All mouse functions can be customized. This manual describes the factory defaults:

A single mouse click with the left mouse button into a text window positions the cursor. Doubleclick to mark the word under the cursor. Hold down the mouse button and drag the mouse pointer over the text to mark the text or - SHIFT button held down - full lines. Columns can be marked by holding down the ALT key while marking or by using the middle mouse button (if your mouse has a middle mouse button).

Drag & Drop

Click at marked text and hold the mouse key down for a moment. This will put the editor into drag and drop mode (the mouse pointer changes from an arrow to a rectangle). You can now move the marked text to the new position by releasing the mouse button over the destination point. Press the STRG key while in drag & drop mode to copy the text (as opposed to moving it).

Maximizing windows

Doubleclick into the status bar of a text window to maximize the window.

Context menus

The right mouse button handling of GoldED deserves special attention: depending on the mouse position the right button either activates the window menu (this is the standard AmigaOS behaviour) or - if the pointer is over a text window - opens a context menu under the mouse pointer. The list of commands offered in
Context Menus
varies depending on the mouse position
and other context details. For example, block-related commands only appear if text has been marked. Context menus can be disabled completely if you prefer standard right mouse button handling (
Configuration/Mouse
)

1.8 Menu

Menu

Project Menu

View Menu

Block Menu

Extras Menu

Layout Menu

Macro Menu

Search Menu

About GoldED

GoldED offers almost unlimited user configuration. There is no ↩ "standard" appearance. Colors, resolution, fonts, menus: all major parts of GoldED are configurable. This manual describes the default menus shipped with the editor. You’ll probably want to customize the standard setup using the

Extras/Customize dialog.

About this manual: On the following pages, references of the form "Project/Open" refer to a function ("Open") listed in a specific menu ("Project"). References of the form "Extras/Customize/Filetypes" refer to a dialog ("Filetypes") opened by the specified menu item ("Customize" in the "Extras" menu).

1.9 Project Menu

Project Menu
Menu tree of project menu

Project/Properties

Project/Save And close

Project/Open

Project/Print

Project/Open In New Window

Project/Use Current path

Project/Open Original

Project/Set Path

Project/Insert File

Project/Clear Document

Project/Append File

Project/Close Window

Project/Save

Project/Iconify

Project/Save As

Project/Exit Editor

Project/Save Compressed File

The project menu offers functions related to the general ↩ handling of a document: load, save, print and similar functions.

1.10 Project/Properties

Project/Properties

Sets the properties of a document: name, comment, protection bits and file type are set by this requester. Below you find a description of the input fields of the properties dialog.

Filetype

Filetypes are the basis of GoldED’s configuration management. A file’s type is determined when you load a document. The result of this check controls the environment assigned to the text (menus, colors, etc). Usually filetype detection is based on the file’s name and the file name suffix. However, you can override automatic detection by providing a virtual name (in the filetype field) which is considered instead of the real file name when the editor determines the type. Example entry: ".c". The editor will then use the contents of the filetype field and treat the text as if it were named ".c", ie. use the environment for files ending on ".c" (the C++ environment if installed).

Document name

The buffer name. This input field sets the window name, it does not rename files on disks (Extras/Rename File can be used to rename files on disk). Changing the name triggers filetype detection and can cause a change in the environment used for editing the file. (see Filetypes).

Protection bits (Readable, Writeable, etc)

The protection bits of the document. These are saved when the document is saved. Please refer to your Amiga manual if you are unfamiliar with the meaning of these bits. Usually only the script bit should be changed: Set the script bit if saving a batch file (a file to be executed with the AmigaDOS command "execute"), clear it for all other text types. The bits are set to a default state as defined by the Configuration/Misc dialog when a document is cleared (Project/Clear Document) .

1.11 Project/Open

Project/Open of: PROJECT MENU

Loads a new document into the current window. The text in that window is discarded (a confirmation dialog has to be answered if that text has been modified). You can select the file to be loaded from a file requester. Both, the ASL file requester and the Reqtools file requester are supported. Choose your preferred requester in the Extras/Customize/Global Options dialog. Compressed files (saved with the Project/Save compressed file function) are decompressed automatically if the XPK libraries are installed on your computer.

Tabs

GoldED recognizes TAB codes (decimal 9) in documents and processes them according to the current TAB settings (as set by the Configuration/Tabs dialog). The TAB settings control how wide tabs appear on screen and whether TAB codes are converted to spaces. Please read the section on Tabulators for a detailed discussion of GoldED’s TAB handling.

Folding

The editor automatically "folds" (see Folding) text sections marked by fold markers when loading documents. Automatic folding can be turned off in the Configuration/Misc dialog.

Loading binary files

GoldED can not handle binary files. Do not attempt to edit binaries (programs). GoldED is a text editor, not a hex editor. It will modify files read into the editor in a way suitable for text files but unsuitable for binaries. For example, it will filter CR codes, possibly expand TAB codes to spaces, clear the executable bit, remove excessive spaces, etc.

Multiselection

Most file requesters presented by GoldED (including the one shown by this menu) support multiple selection: you can select more than one file by holding down the SHIFT key when selecting files.

Drag & Drop

Text windows support drag & drop if GoldED is running on the Workbench screen: You can drag icons of text files over a text window to have the files loaded into the editor. Multiple selection is supported: You can drag multiple files simultaneously over a window by holding down the shift key while selecting icons.

1.12 Project/Open In New Window

Project/Open In New Window of:
PROJECT MENU
Loads a document into a new window (see
Project/Open
).

1.13 Project/Open Original

Project/Open Original of:
PROJECT MENU
Reloads the current file from disk. Useful after you have made ↩
some changes
but want to go back to the unmodified original.

1.14 Project/Insert File

Project/Insert File of:
PROJECT MENU
Inserts a file or multiple files before the cursor line. You can ↩
choose the

file(s) to be inserted from a file requester. Hold down the SHIFT key to select more than one file (multiple selection).

1.15 Project/Append File

Project/Append File of:
PROJECT MENU
Appends one or more files to the current text. You can choose the file(s) to be inserted from a file requester. Hold down the SHIFT key to select more than one file (multiple selection).

1.16 Project/Save

Project/Save of:
PROJECT MENU
Saves the document to the file displayed in the window title. Old copies are overwritten silently in silent overwrite mode (
Configuration/Misc
). A backup of the old version is automatically saved to the global backup path
(
Extras/Customize/Global Options
) if backup creation has been enabled (Configuration/Misc]). The global backup path can be the empty to have backups saved to the same directory the text is saved to (backups are prefixed with "copy of").

Tip: The backup path should be on the partition most of your files are saved to or the performance of the backup function will suffer.

Read-Only-Windows

This function is blocked for read-only windows to prevent you from accidentally overwriting important files (the read-only state is signaled by a ghosted modified flag in the status bar). All reference windows are read-only (see
QuickReference
).
Project/Properties
can be used to set or clear the read-only state. The context menu also lists functions to set or clear the read-only state: press the right mouse button over the modified flag (the field below the window's close button) to see these functions in the context menu.

1.17 Project/Save As

Project/Save As of:
PROJECT MENU
Saves the current buffer to the file you choose from a file requester
(defaults to the name displayed in the window title). Files from this file requester can not be chosen with a doubleclick to prevent accidental selection of the wrong file (you must use the OK button to confirm your choice). This menu function is not available for read-only documents (see

Project/Save
).

1.18 Project/Save Compressed File

Project/Save Compressed File of:
PROJECT MENU
XPK support

Saves the current file in a compressed or encrypted XPK format. See

Configuration/Misc
on how to select a compression or encryption format. XPK encryption formats require a password (set the password with the

Extras/Customize/Global Options
dialog). Encrypted files can not be loaded back into the editor - or other programs - unless the password is set to the password used when saving the file.

This function requires installed XPK libraries. XPK libraries are not shipped with GoldED but can be downloaded from Aminet. Other programs might or might not be able to handle XPK files. Compilers usually can not load XPK files, so saving source codes as compressed XPK files is not such a good idea.

1.19 Project/Save And Close

Project/Save And Close of:
PROJECT MENU
Saves a document and closes its window. The file is not saved but closed if
the window has been declared read-only (see
Project/Save
). GoldED is either
terminated or kept running in the background after the last window has been closed depending on the resident setting (
Extras/Customize/Global Options

).
You can reactivate a resident editor by using the
Hotkey
(right ALT + right
SHIFT + RETURN), by running the quickstarter or by clicking on the Drag &
Drop icon (the Drag & Drop program from the GoldED drawer).

1.20 Project/Print

Project/Print of:
PROJECT MENU
Prints the document. Various printer-related options can be set ↩
with the
printer configuration dialog (
Configuration/Printer
). GoldED's printer
settings override the same fields of the Workbench printer preferences.

Owners of DeskJet and LaserJet printers (or other PCL compatible printers)
should use the printer utility shipped as add-on with GoldED. A printer
symbol appears in the toolbar if the add-on has been installed. The add-on
provides various additional features including doublesided printing, printing
of four or more pages on one page, etc.

1.21 Project/Use Current Path

Project/Use Current Path of:
PROJECT MENU
Instructs the editor to use the path of the current document as ↩
global default
path (see
Project/Set Path
for what this means).

1.22 Project/Set Path

Project/Set Path of:
PROJECT MENU
This function sets the global default path. You can choose a path ↩
from a path
requester.

The initial value of the global default path is the path of the program which
has been used to start GoldED. For example, if you set the path of a shell

window to RAM: (with a "CD RAM:" command), GoldED will inherit this path if started from the same shell window.

The global default path is considered by various functions of GoldED:

Project/Clear Document resets the document name to point to the default path and the Project/Open In New Window function does show it as default in the file requester. Setting the global default path doesn't change the name of the current document (use Project/Properties instead).

1.23 Project/Clear Document

Project/Clear Document of:
PROJECT MENU

Clears the current window. You must answer a confirmation ↩ dialog if the document has been modified. Protection bits are reset to the defaults as defined by Configuration/Misc. The path is reset to the global default path (see Project/Set Path) and the file name is reset to "unnamed" (plus an optional suffix).

1.24 Project/Close Window

Project/Close Window of:
PROJECT MENU

Discards the current text and closes the window. You must ↩ answer a confirmation dialog if the document has been modified. GoldED is either terminated or kept resident in the background depending on the resident setting (Extras/Customize/Global Options). You can reactivate a resident editor by using the hotkey (right ALT + right SHIFT + RETURN), by running the quickstarter or by clicking on the Drag & Drop icon (the Drag & Drop program from the GoldED drawer). The obvious advantage of keeping the editor resident is the reduced startup time. Main disadvantage is increased memory consumption.

GoldED internally uses an asynchronous memory management: you won't have to wait until all memory has been freed when you close a window. Your Amiga will however feel slightly slower than usual while the background task is busy with freeing memory - especially if the text buffer has been huge.

1.25 Project/Iconify

Project/Iconify of:
PROJECT MENU
Hides all windows. The documents are not discarded but added to ↩ the list of hidden buffers (see
View/Hidden Documents
). A small GoldED icon is created on the Workbench screen. Doubleclick on this icon to reactivate the editor, ie. to open the first buffer from the list of hidden buffers in a new window.

1.26 Project/Exit Editor

Project/Exit Editor of:
PROJECT MENU
Closes all text buffers and terminates GoldED. You must answer a ↩ confirmation dialog if documents have been modified. The editor is completely removed from memory and is not kept running in the background (see
Project/Close Window
) unless a
Debugger
has locked GoldED: GoldED can not be terminated during a debugger session.

1.27 Block Menu

Block Menu
Menu tree of block menu

Block/Mark Text

Block/Block Delete

Block/Mark Lines

Block/Block Move

Block/Mark Columns

Block/Block Copy

Block/Mark Paragraph

Block/Edit Block

Block/Mark All

Block/Convert To Uppercase

Block/Marker Off

Block/Convert To Lowercase

Block/Clipboard Cut

Block/Print Block

Block/Clipboard Copy

Block/Save Block

Block/Clipboard Paste

All functions of the block menu are related to the management of ↩ blocks which are marked sections of text. Use the menu functions or the Mouse to mark text. Three types of block markers (arbitrary text, full lines, columns) are available and can be selected by holding down a qualifier key (SHIFT or ALT) when marking with the mouse. Every document can have its own block. When marking with the mouse, the character under the cursor is either included in the block or excluded from the block depending on the user interface configuration (Configuration/User Interface).

1.28 Block/Mark Text

Block/Mark Text of:
BLOCK MENU

This function starts the ink flow for marking text. Move the ↩ cursor over the text to be marked and use this function again to stop the ink flow. Use

Block/Marker Off
to turn the marker off when no longer needed. Cursor movements either set the new block start or the new block end depending on the current cursor position (the block start is set if the cursor is closer to the block start than to the block end and vice versa).

Some functions of this editor require line blocks. For example, you can't mark a single word and block-format it. GoldED automatically converts arbitrary blocks to line blocks on an as-needed basis.

1.29 Block/Mark Lines

Block/Mark Lines of:
BLOCK MENU
Sets the beginning or end of a block of lines depending on whether ↩
the cursor
is closer to the current block start or closer to the block end. Use this command to quickly mark whole lines. See
Block/Mark Text
or
Mouse
on how
to mark arbitrary text. Use
Block/Marker Off
to turn the marker off.

1.30 Block/Mark Columns

Block/Mark Columns of:
BLOCK MENU
This function starts the ink flow for marking text columns. Move ↩
the cursor
over the text to be marked and use this function again to stop the ink flow. You can use the mouse to mark columns, too, by holding down the ALT key while marking or by using the middle mouse button (if your mouse has a middle mouse button). The menu functions for moving and copying text will copy (or move) columns in "overlay" mode: the column is mixed to the existing text (as opposed to inserting new lines). This feature is meant to simplify creation of text with multiple columns.

1.31 Block/Mark Paragraph

Block/Mark Paragraph of:
BLOCK MENU
Marks a paragraph: all lines between the last empty line and the ↩
next empty
line are marked. Use this function again (or
Block/Marker Off
) to hide the

selection.

1.32 Block/Mark All

Block/Mark All of:
BLOCK MENU
Marks all lines of a text.

1.33 Block/Marker Off

Block/Marker Off of:
BLOCK MENU
Hides the block marker.

1.34 Block/Clipboard Cut

Block/Clipboard Cut of:
BLOCK MENU
Cuts the marked text (see
Block/Mark Text
or
Mouse
) and saves it 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 window using
Block/Clipboard Paste
).

You shouldn't use this function to move or copy data within a GoldED document:
Block/Block Copy
or
Block/Block Move
perform the same task much faster and without saving data to the clipboard (ie. without consuming additional memory).

Clipboard

The clipboard is used by applications to exchange data. It offers several storages called "units" and thus can hold several blocks of data simultaneously. GoldED can access any unit (see
CLIP

```
                 command)  but  usually
unit  0 is used to share data between applications. Postings to the clipboard
are formatted as IFF files. GoldED supports IFF/FTXT clipboard  access  (text
files).  Other  files posted to the clipboard (e.g. images in IFF format) can
not be insert into editor windows. Example: mark  text  in  a  shell  window,
press  AMIGA-C to copy the text to the clipboard (unit 0) and insert the clip
into a GoldED window by using the AMIGA-V shortcut. A paste  operation  won't
remove the data from the clipboard, i.e. you can paste the same data multiple
times.
```

1.35 Block/Clipboard Copy

```
                Block/Clipboard Copy of:
                 BLOCK MENU
                Saves the marked text (see
                 Block/Mark Text
                 or
                 Mouse
                )  to  a
                 Clipboard
                unit, ready to be inserted into applications supporting the  ←
                   clipboard device.
```

1.36 Block/Clipboard Paste

```
                Block/Clipboard Paste of:
                 BLOCK MENU
                Inserts the clipboard contents  into  the  text.  Depending  on   ←
                   the  current
insertion mode (
                 Layout/Insert mode
                ) pasted text is either inserted into the
existing text or overwrites existing text. You will see a  'clipboard  error'
if the clipboard is empty.

Paste Unit

  Inserts the contents of a
                 Clipboard
                 unit into the  current  text.  Single
  words  found  in  the clipboard are inserted at the current cursor position
  but paragraphs (multiple lines) are treated differently: they are  inserted
  before the current line.

Paste Unit As Column

  Inserts the
                 Clipboard
```

contents as a text column at the current cursor position. The text read from the clipboard is mixed to the existing lines (overlay) so that you can use this function to create text with multiple columns.

1.37 Block/Block Delete

Block/Block Delete of:
BLOCK MENU
Deletes the marked text (see
Block/Mark Text
or
Mouse
).

1.38 Block/Block Move

Block/Block Move of:
BLOCK MENU
Moves the marked text text (see
Block/Mark Text
or
Mouse
) to the cursor position (if single words have been marked) or to the cursor line (if lines have been marked).

This function does not use the slow
Clipboard
device. While it is very useful to duplicate sections within one editor window it can not be used to move text from one window to another window or to share data with other applications. Use
Block/Clipboard Cut
and
Block/Clipboard Paste
for those tasks.

1.39 Block/Block Copy

Block/Block Copy of:
BLOCK MENU
Copies the marked text (see

Block/Mark Text
or
Mouse
) to the cursor
position.

This function does not use the slow clipboard device. While it is very useful to duplicate sections within one editor window it can not be used to copy text from one window to another window or to share data with other applications. Use
Block/Clipboard Cut
and
Block/Clipboard Paste
for those
tasks.

1.40 Block/Edit Block

Block/Edit Block of:
BLOCK MENU
APPEND TEXT

Appends a text to all marked lines. Example: Mark some lines and use 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 ";".

INSERT TEXT COLUMN

Inserts a text string into all marked lines. Example: Mark some lines, move the cursor to the destination column (e.g. column 3) and call this function. A requester pops up, asking you for a text to be inserted. Enter 'Prototype'. GoldED inserts this text into all marked lines at column 3. This function simplifies the creation of tables: insert '|' to create a vertical line.

DELETE COLUMN

Deletes the cursor column from all selected lines. Move the cursor to the column to be removed before you use this function (e.g. move cursor to column 10 if you want to delete column 10 from all block lines). You shouldn't use this function to change the indention of a paragraph (to avoid accidentally deleting text): Use
Layout/Indent Block
instead.

INSERT COLUMN

Inserts a new (empty) column into all marked lines. Move the cursor to the correct position before you use this function. Example: Move the cursor to column 40 if you want to insert one empty column at position 40 into all block lines.

SORT (COMPARE AT CURSOR POSITION)

Sorts the marked lines alphabetically. This function is case-insensitive (i.e. 'A' and 'a' are considered equal). Comparisons start in the cursor column. Text before this column is ignored for the comparison but included when sorting the lines.

SORT (COMPARE AT START OF LINES)

Sorts the marked lines alphabetically. This function is case-insensitive (i.e. 'A' and 'a' are considered equal). Comparisons start in the first column of each block line.

1.41 Block/Convert To Uppercase

Block/Convert To Uppercase of:
BLOCK MENU
Converts the marked text to uppercase characters. This function ↩ uses the
locale library to convert non-ASCII characters (e.g. 'ß') correctly.

1.42 Block/Convert To Lowercase

Block/Convert To Lowercase of:
BLOCK MENU
Converts the marked text to lowercase characters. This function ↩ uses the
locale library to convert non-ASCII characters (e.g. 'ß') correctly.

1.43 Block/Print Block

Block/Print Block of:
BLOCK MENU
Prints the marked text. The standard preferences printer is ↩ used. See

Configuration/Printer
on how to customize the printer settings.

1.44 Block/Save Block

Block/Save Block of:
BLOCK MENU
Saves the marked text (see
Project/Save
).

1.45 Layout Menu

Layout Menu
Menu tree of layout menu

Layout/Set Right Border

Layout/Format Paragraph Block

Layout/Keep Indention

Layout/Format Paragraph Left

Layout/Insert Mode

Layout/Format Paragraph Center

Layout/Word Wrap

Layout/Format Paragraph Right

Layout/Correct Case

Layout/Format Block Block

Layout/Detect Templates

Layout/Format Block Left

Layout/Tabs To Spaces

Layout/Format Block Center

Layout/Indent Line

Layout/Format Block Right

Layout/Indent Block
All functions of the layout menu are related to formatting a text.

Paragraph vs. block

Some formatting functions modify the marked lines (see

Block/Mark Text
)
while others modify the paragraph the cursor is in (the lines between the last and the next empty line) and do not require that you mark text before you use those functions.

1.46 Layout/Set Right Border

Layout/Set Right Border of:
LAYOUT MENU
Sets the right margin to the column the cursor is in. Margin ↩ settings are
respected by the formatting functions only (e.g.
Word Wrap
). Layout margins
do not restrict cursor movements.

1.47 Layout/Keep Indention

Layout/Keep Indention of:
LAYOUT MENU
Formatting functions (including word wrap) ignore the left margin ↩ defined in
the layout configuration if this option is enabled. Instead, the left margin is set to the indention of the cursor line. This option simplifies reformatting a text with a lot of different indentions.

1.48 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 (OVER = Overwrite).

1.49 Layout/Word Wrap

Layout/Word Wrap of:
LAYOUT MENU
Word Wrap

Toggles word warp mode on or off. The current status is displayed in the screen title (WRAP = Word Wrap). The editor automatically inserts a linefeed if the cursor is moved beyond the right margin while typing with Word Wrap enabled: Words exceeding the right margin are moved to the next line.

1.50 Layout/Correct Case

Layout/Correct Case of:
LAYOUT MENU
Toggles automatic case correction (
AutoCase
) on or off. Case correction is based on a user-defined dictionary (
Configuration/Dictionary
). The case of known words is checked (and possibly corrected) when the cursor leaves a line you have modified.

1.51 Layout/Detect Templates

Layout/Detect Templates of:
LAYOUT MENU
Templates

Enables (or disables) template detection. The current status is displayed in screen's title bar (TMPL = Templates). The editor will look for user-defined patterns (templates) in the input stream if this option is enabled. Templates detected in the input stream are replaced by the "action" (usually a text) associated with the template. For example, you could configure the editor to automatically replace "snc" in the input stream with "sincerely" by defining the template "snc" and configure it to add "sincerely" if detected. See

Configuration/Templates
on how to add or edit templates.

1.52 Layout/Tabs To Spaces

Layout/Tabs To Spaces of:
LAYOUT MENU

Converts all TAB codes (see
Tabulators
) in a document to spaces.

1.53 Layout/Indent Line

Layout/Indent Line of:
LAYOUT MENU
Modifies the indention of the cursor line. A similar indention functions is
assigned to the cursor keys (see
Cursor Keys
) when used together with the
CTRL+ALT keys.

1.54 Layout/Indent Block

Layout/Indent Block of:
LAYOUT MENU
Shifting

Changes the indention of the marked lines (
Block/Mark Text
). A similar
indention functions is assigned to the cursor keys (see
Cursor Keys
) when
used together with the CTRL key.

1.55 Layout/Format Paragraph

Layout/Format Paragraph of:
LAYOUT MENU
Layout/Format Paragraph Block
Layout/Format Paragraph Left
Layout/Format Paragraph Right
Layout/Format Paragraph Center

Reformats the paragraph the cursor is in (see
Paragraph vs. block
). Use

Configuration/Layout
to set the layout margins.

1.56 Layout/Format Block

Layout/Format Block of:
LAYOUT MENU
Layout/Format Block Block
Layout/Format Block Left
Layout/Format Block Right
Layout/Format Block Center

Reformats the marked Lines (see
Block/Mark Text
). Empty lines are not
removed during formatting, your document's basic structure remains unchanged
(but excessive spaces are removed). Use
Configuration/Layout
to set the
layout margins.

1.57 Search Menu

Search Menu
Menu tree of search menu

Search/Search

Search/Reference...

Search/Search Next

Search/Find in Files

Search/Search Backwards

Search/Show Matching Bracket

Search/Search With Index

Search/Check Nesting

Search/Replace

Search/Find File

Search/Replace Next

Search/Find File...

Search/Count Pattern

Search/Show Function

Search/Reference

Search/Show Function List

1.58 Search/Search

Search/Search of:
SEARCH MENU
Shows a requester to enter the text to search for.

Only Whole Words

Enable this option to have substrings which to match the search pattern ignored: The search function will search full words only (delimited by a white-space character).

Wildcards

GoldED supports AmigaDOS wildcards if you enable the "wildcards" option in the search requester: The reserved AmigaDOS wildcard characters (*, |, [] and &) are interpreted in wildcard mode instead of being searched literally. Wildcard search is line-oriented: Enter "Prototype*" as search pattern to search the next line beginning with "Prototype". Enter "*Prototype*" to find lines containing the text "Prototype".

History

Use the arrow button in the search requester to see a list of recent search strings.

1.59 Search/Search Next

Search/Search Next of:
SEARCH MENU
Searches the next occurrence of the search pattern (see
Search/Search
).

1.60 Search/Search Backwards

Search/Search Backwards of:
SEARCH MENU
Searches the last occurrence of the search pattern (see
Search/Search
)
before the cursor line.

1.61 Search/Search With Index

Search/Search With Index of:
SEARCH MENU
Lists all lines containing the search pattern you ↩
specify. See

Search/Search
for a description of the search options.

1.62 Search/Replace

Search/Replace of:
SEARCH MENU
Displays a requester to enter the text to be searched and a ↩
replacement for
this text (can be empty to have the search pattern removed completely from the document). Use the BLOCK button if you want the replacement operation to be restricted to the marked text (
Block/Mark Text
). Please refer to

Search/Search
for a description of search options.

1.63 Search/Replace Next

Search/Replace Next of:
SEARCH MENU
Replaces the next occurrence of the search pattern with the ↩
replacement text.
Search pattern and replacement are set by the
Search/Replace
dialog.

1.64 Search/Count Pattern

Search/Count Pattern of:
SEARCH MENU
Shows a requester to enter the text to be counted. Please ↩ refer to

Search/Search
for a description of search options.

1.65 Search/Reference

Search/Reference of:
SEARCH MENU
QuickReference

Shows a help text related to the word the cursor is placed over (if available). Example: Type 'struct RastPort', move the cursor over 'RastPort' and use this function. A new text window should pop up, showing you the file 'graphics.h' from you compiler's include directory. The cursor will be placed in the first line of the structure definition of '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 them.

Setting up the reference system

Set up the reference system before you use this command (i.e. tell the editor where to look for files). Use the
Configuration/Reference files
menu to
configure the reference system: Simply add the files and directories to be referenced to the list of reference files. The editor will scan these files and create a database of known expressions. C-sources, include files, autodocs, BASIC programs, assembler sources and Pascal sources are supported. Scanners for other filetypes can be added (see
Search/Show Function List
)
so that GoldED's capability to extract reference information from files can be extended to handle virtually all filetypes

1.66 Search/Reference...

Search/Reference... of:
SEARCH MENU
Shows a dialog box to look up an expression using GoldED's ↩ reference system
(see

Search/Reference
). Example: Enter "RastPort" to have the structure definition of "struct RastPort" displayed. This feature can not be used until the reference system has been set up correctly.

1.67 Search/Find In Files

Search/Find In Files of:
SEARCH MENU
Global search

Searches a string in a list of files. File names are highlighted if the pattern is found in the file (SHIFT-click on a file name to toggle highlighting). Doubleclick on a file name to load the corresponding file. The "touch" button updates the creation date of all highlighted files. This is useful to have these files recompiled: most compilers will think that the file has been modified and recompile it during the next make process.

1.68 Search/Show Matching Bracket

Search/Show Matching Bracket of:
SEARCH MENU
Moves the cursor over the matching bracket. This function ↩ handles several bracket types: (), <>, etc. Example: move the cursor over the first (open) parenthesis of a function in a C source code, then use this function to find the matching parenthesis. The editor might not be able to find the matching bracket if confused by brackets in strings and comments (as it does not understand the syntax of the programming language you are using).

1.69 Search/Check Nesting

Search/Check Nesting of:
SEARCH MENU
Validates the correct use (nesting) of "(" and ")" in the current ↩ line. You will see a warning if the number of opening braces does not match the number of closing braces. The editor might not be able to correctly count the braces if confused by brackets in strings and comments (though it does try to ignore braces in strings).

1.70 Search/Find File

Search/Find File of:
SEARCH MENU
Move the cursor over a file name and use this function to find ↩ the file on
your computer. You have to configure the file search feature (ie. specify the places to be searched) before you can use this function. See

Configuration/File Search
for details.

1.71 Search/Find File...

Search/Find File... of:
SEARCH MENU
Shows a dialog box where you can enter the name of a file to be ↩ searched on
your computer. You have to configure the file search feature before you can use this function. See
Configuration/File Search
for details.

1.72 Search/Show Function

Search/Show Function of:
SEARCH MENU
Moves the cursor to the definition of a function (if found in the ↩ same file).
Place the cursor over the function to be located before you use this menu. Internally, an invisible list of functions in the current document is created to locate the function you are looking for. The creation of this list is handled by a QuickFunc scanner (see
QuickFunc
for details).

1.73 Search/Show Function List

Search/Show Function List of:
SEARCH MENU
QuickFunc

Shows a list of all functions (or other data structures) found in the document. Click at a list entry to move the cursor to the function definition. Recognized structures depend on the document's type and the scanner associated with the type.

Configuration

Use the "mode" button below the list of recognized elements to choose a specific scanner (or to add a new scanner). The list of scanners is priority-sorted: the editor will use the first scanner capable of handling the document when choosing a scanner (the file type pattern associated with the scanner must match the document name). The scanner configuration does affect the reference system (
Search/Reference
), too, because these scanners
are used by GoldED when creating reference databases.

Technical information

You can program your own scanners. Scanners are LoadSeg()'ed by GoldED, i.e. are expected to be executables. Example source code is available in the golded:developer/examples/scanner drawer. A scanner is called once for every line of a document. It will receive the address of the text (char **) in A0. The length is passed in D0, the line number is passed 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 scanners point of view (a scan handler looking for #defines returns NULL if the line doesn't contain a #define). It is expected to return the length of a result string if it does find interesting information in the line (and to set the string pointer in A0 to point to the result string). For example, a scanner for #defines would return the name of the defined constant.

1.74 View Menu

View Menu
Menu tree of view menu

View/Open New Window

View/Show Plug-Ins

View/Go To Line

View/Show Breakpoints

View/Go To Offset

View/Show Colors

View/Go To Beginning/End

View/Show Preview

View/Go To Modification

View/Windows

View/Store Position

View/Activate Next Window

View/Recall Position

View/Activate Prev Window

View/Folding

View/Hidden Documents

View/Show Toolbars

1.75 View/Open New Window

View/Open New Window of:
VIEW MENU
Opens a new text window. Window size and window position are determined by the local configuration (Configuration/Display): the window is centered on the screen, aligned with other windows or maximized. Use View/Windows/Remember Dimensions to set the default window size for new windows.

1.76 View/Go To Line

View/Go To Line of:
VIEW MENU
Moves the cursor to the line you specify (the first line in a text is line 1). Enable the ’unfold’ option in this dialog box to have folded (invisible) lines considered.

1.77 View/Go To Offset

View/Go To Offset of:
VIEW MENU
Moves the cursor to the byte offset you specify (the offset of the first
character in a text is 1).

1.78 View/Go To Beginning/End

View/Go To Beginning/End 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 line if it is closer to the text end than to
the text start (and vice versa).

1.79 View/Go To Modification

View/Go To Modification of:
VIEW MENU
Moves the cursor to the position of the last change.

1.80 View/Store Position

View/Store Position of:
VIEW MENU
Records the position of the cursor. Use
View/Recall Position
to restore the
cursor position. You can store up to five positions for each text window. The
position markers will move with the text when text is entered or deleted.

1.81 View/Recall Position

View/Recall Position of:
VIEW MENU
Recalls a cursor position stored by
View/Store Position
.

1.82 View/Folding

View/Folding of:
VIEW MENU
Folding

View/Folding/Fold or Unfold
View/Folding/Fold all
View/Folding/Unfold all

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. Nested folding is supported: folds may contain further folds.

FOLD

How to fold lines

Enclose the section to be folded into 'fold markers'. Fold markers are short character sequences. The default marker 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 and a single new line (the fold header) appears:

```
> important function
```

Unfolding

Press the HELP key again to unfold the fold. The 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 can decrease the performance.

Fold markers may not be longer than ten characters. The shorter the faster. Nested folding is supported if different (!) markers for beginning and end of folds are used (see
Configuration/Misc
). You should choose fold

markers carefully. 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 blocked if the cursor is over a fold header. Unfold a fold if you want to edit it or search text in it. Block operations (
Block Menu
) don't modify folded sections, but 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 the text for fold markers. Lines enclosed by fold markers are folded. There is a keyboard shortcut for this function: CTRL + HELP. It toggles all folds, ie. it unfolds all folds if the cursor is placed over a fold header (but folds all folds if the cursor is placed over normal text).

UNFOLD ALL

All folds (and sub-folds) are unfolded.

1.83 View/Show Toolbars

View/Show Toolbars

Shows or hides the toolbars. Look and feel of the toolbar and the functions assigned to the buttons are configurable (
Configuration/Toolbars
).

1.84 View/Show Plug-Ins

View/Show Plug-Ins

Shows or hides the plug-in containers (
Configuration/Plug-Ins
).

1.85 View/Show Breakpoints

View/Show Breakpoints of:
VIEW MENU
Debugger

Enables or disables the breakpoint display of the current window. This function is useless unless you have a source level debugger with built-in support for GoldED (e.g. the StormC debugger). Usually the debugger controls the breakpoint display automatically. This menu permits you to override debugger control. Breakpoints are created by the compiler or by the debugger. The user can set breakpoints to "active" or "inactive" or delete breakpoints (adding new breakpoints doesn't make sense). Click at a breakpoint to toggle its state. Breakpoints can be deleted with a SHIFT-click.

A typical debugger session using GoldED as debugger frontend could proceed like this: Compiler and debugger are started by the compiler environment. The debugger assumes control over GoldED if it has been prepared for GoldED. Breakpoints appear automatically in all text windows connected to your project. You'll use the debugger window to step through the program (or to 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, ie. you can correct errors immediately. Unmodified buffers are simulated towards the debugger to avoid confusing it with a modified text.

Information about breakpoints optionally is saved to files (see

Configuration/Misc
) so that you can restore breakpoints quickly to a specific state during the next debugger session: The editor saves breakpoint information to files named like the original text but with the suffix set to ".bpt". These files are created automatically when you save a text containing breakpoints and loaded automatically when loading the text. They are deleted automatically if you save the text without breakpoints. Please note that it depends on your debugger whether 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.

1.86 View/Show Colors

View/Show Colors of:
VIEW MENU
Enables or disables color coded text (
Configuration/Syntax Highlighting
).

1.87 View/Show Preview

View/Show Preview of:
VIEW MENU
Preeview

Redisplays the current text with the alternate font (the preview font). Use this function again to return to the normal font.

1.88 View/Windows

View/Windows

This menu offers window-oriented functions:

View/Windows/Remember Dimensions
View/Windows/Arrange Vertically
View/Windows/Arrange Horizontally
View/Windows/Window Zip
View/Windows/Window Maximize
View/Windows/Window Center
View/Windows/Activate Dialog Box

REMEMBER DIMENSIONS

Size and position of the current window are stored as default window dimensions for new windows sharing the display configuration of the current text. Note that other configurable options of GoldED can override the default dimensions (e.g.
Configuration/Display/Details
).

ARRANGE VERTICALLY, ARRANGE HORIZONTALLY

Arranges the windows on a screen to avoid overlapping windows. The screen's title bar is kept free (correct recognition of the title bar is not possible if the screen has been dragged down). You can assign an extra weight to the current window using the "priority" field of

Configuration/Display
. This function recognizes autoscroll screens (screens larger then the display): the window(s) are arranged in the visible rectangle only. Screen areas to be kept free (e.g. for a Toolmanager toolbar) can be specified (
Configuration/Display
).

WINDOW ZIP

Resizes the current window to the alternate dimensions (as if using the window's zip button). The alternate window size is preset by GoldED to the minimum window size the editor can handle.

WINDOW MAXIMIZE

Enlarges the current window to cover the visible display area. User defined margins and the position of the screen’s title bar are considered.

WINDOW CENTER

Centers the current window on the screen. User defined margins and the position of the screen’s title bar are considered.

ACTIVATE NEXT DIALOG BOX

Activates the next open dialog box so that you can quickly activate another window with your keyboard (use the
Hotkey
to activate the first text
window).

1.89 View/Activate Next Window

View/Activate Next Window of:
VIEW MENU
Activates the next text window (so that you don’t have to grab the mouse for
window activation). Windows are sorted chronologically.

1.90 View/Activate Prev Window

View/Activate Prev Window of:
VIEW MENU
Activates the previous text window (so that you don’t have to grab the mouse
for window activation). Windows are sorted chronologically.

1.91 View/Hidden Documents

View/Hidden Documents of:
VIEW MENU
View/Hidden Documents/Hide Window
View/Hidden Documents/Show Document
View/Hidden Documents/Swap With
View/Hidden Documents/Swap With Next Window
View/Hidden Documents/Swap With Prev Window

HIDE WINDOW

Hides a text buffer, i.e. closes the window but keeps the document. Useful to have several text files present without cluttering the display. Use "Show Document" 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 use the commodities exchange program of your Workbench.

SHOW DOCUMENT

Asks you for a hidden buffer to be reopened.

SWAP WITH

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

SWAP WITH NEXT WINDOW

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.

1.92 Extras Menu

Extras Menu

Menu tree of extras menu

Extras/Document Statistics

Extras/Line Duplicate

Extras/Undo

Extras/Line Remove

Extras/Redo

Extras/Line Insert

Extras/Insert Text

Extras/Line Execute

Extras/Complete Text

Extras/Open Shell

Extras/Insert Special Character

Extras/Delete File

Extras/ASCII Code

Extras/Rename File

Extras/Swap Adjacent Lines

Extras/Customize

1.93 Extras/Document Statistics

Extras/Document Statistics of:
PROJECT MENU
Shows statistics for the current text: Bytes, lines, ↩ folds and non-ASCII-characters (codes above 127) are counted and the width of the longest line is determined. Invisible linefeed characters are included in the figures. Additionally, the total UNDO RAM consumption (of all texts) is displayed: This figure should help you to configure the undo buffer size (
Extras/Customize/Global options
).

1.94 Extras/Undo

Extras/Undo of:
EXTRAS MENU
Undo & Redo

Reverts the last operation (if the undo/redo feature has been enabled; see (
Extras/Customize/Global options
). You can undo an undo by using

Extras/Redo
immediately (you won't be able to redo undos if you modify the text inbetween). The number of undoable operations depends on how much memory you have assigned to the undo mechanism. All operations including block-related functions, formatting and loading can be undone. However, depending on the selected undo mode, some operations can not be undone separately. In standard mode, all modifications within a single line count as one undo step and can only be undone together. In "High" mode, all major changes (e.g. "delete until end of line") within a line can be undone

separately while small changes (single keystrokes) are collected and count as one undo step. In "Auto" mode, all modifications can be undone separately. A garbage collection has been added to reduce the memory consumption in "Auto" mode: multiple old small operations (keystrokes) are automatically converted to one undo step after a while. Memory consumption created by he undo system can be checked with the
Extras/Document Statistics
menu.

1.95 Extras/Redo

Extras/Redo of:
EXTRAS MENU
Reverts the last undo (
Extras/Undo
). This command can only be used immediately after an undo: All redo information is discarded once you start editing the text.

1.96 Extras/Insert text

Extras/Insert text of:
EXTRAS MENU
Extras/Insert text/Insert Path To File
Extras/Insert text/Insert Path To Folder
Extras/Insert text/Insert Date
Extras/Insert text/Insert Time

INSERT PATH TO FILE, INSERT PATH TO FOLDER

Inserts a file name or a path name (chosen from a file requester) into the document.

INSERT DATE

Inserts the current date into the text. You should set the environment variable USERTOWN to the name of your town before using this menu. The AmigaDOS command for setting environment variables is setenv. Example: setenv USERTOWN "Boston". The setenv commands saves variables to the temporary "env:" directory only. Copy "env:usertown" to the permanent directory "envarc:" to have it saved permanently. The date format can be set with
Extras/Customize/Global Options
.

INSERT TIME

Inserts the current time into the document.

1.97 Extras/Complete Text

```
                Extras/Complete Text of:
                 SEARCH MENU
                Extras/Complete Text/Complete With Dictionary
Extras/Complete Text/Complete With Context

COMPLETE WITH DICTIONARY

  Replaces the word under the cursor with a word  from  the  dictionary.  See

                 APC
                 for details.

COMPLETE WITH CONTEXT (INTELLISENSE)

  Replaces the word under the cursor with a word taken from the context,  ie.
  with  a  "similar"  word  from the document. Example: Place the cursor over
  "rast" and use this menu or press SHIFT-SPACE: "rast"  is  expanded  to  an
  expression  starting  with  "rast"  found  in  the  same document (possible
  "RastPort" if you are a C programmer). This function obviously makes  sense
  only if you can predict the expanded form (which is not very difficult with
  some experience).
```

1.98 Extras/Insert Special Character

```
                Extras/Insert Special Character of:
                 EXTRAS MENU
                Shows a table with all characters available in the current text   ↩
                   font.  Click
at a character to have it inserted into the document.
```

1.99 Extras/ASCII Code

```
                Extras/ASCII Code of:
                 EXTRAS MENU
                Search/ASCII Code/Show ASCII Code
Search/ASCII Code/Insert ASCII Code
Search/ASCII Code/Insert Code ESC
Search/ASCII Code/Insert Code FF
Search/ASCII Code/Convert Text
Search/ASCII Code/Remove CR
Search/ASCII Code/Convert To Uppercase
Search/ASCII Code/Convert To Lowercase
```

SHOW ASCII CODE

Shows the ASCII code of the character under the cursor.

INSERT ASCII CODE

Inserts the ASCII code (0-255) you specify into the text.

INSERT CODE ESC, INSERT CODE FF

Inserts the specified special ASCII code into the text: ESC = code 27 (Escape) or FF = code 12 (Formfeed).

CONVERT TEXT

Character Set Remapping

Remaps the text to another character set. Remapping means that each character is replaced by another character defined in the translation file you choose in the dialog box. The result depends on the translation file. Example: the text is remapped to the MS-DOS format if you choose the 'AmigaToMSDOS' translation file. Without this conversion some characters would have been displayed incorrectly if the file were transfered to a PC. Remapping might be lossy. Example: the "©" character can not be mapped to a similar MSDOS character (there is no such character in the PC character set). "c" is used instead.

REMOVE CR

All CR codes are removed from the document by this command. Files transferred from the PC to the Amiga have lines ending on CR+LF while files on the Amiga only have an LF code at the end of each line. The (superfluous) CR code shows up as inverted "C".

CONVERT TO UPPERCASE, CONVERT TO LOWERCASE

Converts the character under the cursor to uppercase (or lowercase).

1.100 Extras/Swap Adjacent Lines

Extras/Swap Adjacent Lines of:
EXTRAS MENU
Swaps the cursor line with the next line.

1.101 Extras/Line Duplicate

Extras/Line Duplicate of:
EXTRAS MENU
Duplicates the cursor line.

1.102 Extras/Line Remove

Extras/Line Remove of:
EXTRAS MENU
Deletes the cursor line. The line is put to a pick-push ring ↩
buffer (last in,
first out). You can reinsert it with the
Extras/Line Insert
menu. Example:
Delete 3 lines, move the cursor to another line and use
Extras/Line Insert
three times to reinsert the lines. The pick/push buffer holds up ↩
up 50 lines.
This function is assigned to the keyboard shortcut CTRL-DEL. Keyboard access
(CTRL-DEL) will give you a much better response time than the menu due to the
AmigaOS input queue handling (there is actually none for menu hotkeys).

1.103 Extras/Line Insert

Extras/Line Insert of:
EXTRAS MENU
Reinserts the last line from the pick/push buffer. See
Extras/Line Remove
for details.

1.104 Extras/Line Execute

Extras/Line Execute of:
EXTRAS MENU
Executes the cursor line as DOS command. Useful to embed compiler ↩
calls (in
comments) into source codes. This 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
  ------------------------------------------------------------------
*/
```

1.105 Extras/Open Shell

```
                Extras/Open Shell of:
                 EXTRAS MENU
                Opens a shell window on GoldED’s screen.
```

1.106 Extras/Delete File

```
                Extras/Delete File of:
                 EXTRAS MENU
                Deletes the file(s) you select from a file requester. It is not   ←
                   possible  to
delete delete-protected files.
```

1.107 Extras/Rename File

```
                Extras/Rename File of:
                 EXTRAS MENU
                Renames the file(s) you select from a file requester.
```

1.108 Extras/Customize

```
                Extras/Customize of:
                 EXTRAS MENU
                The dialog opened by this menu is the main configuration window of ←
                    GoldED. It
is  used to set the global options and to configure the filetypes (while this
requester can be used to configure the environment of the  current  document,
too,  a  faster  method of accessing the local
                 configuration
                 is the context
menu).

The dialog has three pages:
```

Extras/Customize/Information

Extras/Customize/Global Options

Extras/Customize/Filetypes

1.109 Extras/Customize/Information

Extras/Customize/Information of:
EXTRAS MENU
General GoldED information page (version, copyright). The ↩
Connect-To-WWW
button on this page can only be used if you have installed the RA package shipped with GoldED. Please connect to the Internet with your browser before using that button.

1.110 Extras/Customize/Global Options

Extras/Customize/Global Options of:
EXTRAS MENU
Global configuration. Global options affect all open documents.

Extras/Customize/Global Options/Variables
Extras/Customize/Global Options/Undo
Extras/Customize/Global Options/Options
Extras/Customize/Global Options/Misc

Extras/Customize/Global Options/Variables

Configuration dialog for the user-defined variables: you can 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
(
Variables
). They can be used to control the state of menu checkmarks
(
Configuration/Menus
), too. The purpose of user-defined variables is to support Rexx macro programmers: The main advantage compared to Rexx variables is that variable values are saved to the configuration.

Extras/Customize/Global Options/Undo

UNDO MODE

Enables or disables the undo mechanism (Undo & Redo) and selects an undo mode: Standard mode, high mode and auto mode are available. In "Standard" mode you can undo and redo major changes only. For example, you can undo formatting operations. Changes within lines can not be undone separately. In "High" mode major changes within lines can be undone separately though you still won't be able to undo single keystrokes separately. In "Auto" mode all changes (including single keystrokes) can be done separately (see Extras/Undo).

UNDO BYTES, UNDO STEPS

The editor backups text you are about to change or delete if the undo feature is enabled. 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 in 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 editor's performance. Avoid sizes below 100K. The undo buffer size is understood as suggestion. Short time memory usage might exceed the limit. For example, buffer overflows created by Extras/Undo are tolerated to permit a redo Extras/Redo .

Extras/Customize/Global Options/Options

EDITOR RESIDENT

GoldED supports Hotkey activation if this option is enabled:

Hotkey

The editor is not completely removed from memory after the last window has been closed. Instead, it will wait in the background for you to press the hotkey combination: right SHIFT & right ALT & RETURN. The commodities exchange program of your Workbench can be used to reactivate or unload GoldED, too. Hotkey activation gives you a quick response time (the editor appears almost instantly) at the cost of increased memory consumption. You should disable this feature on systems with less than 3 MB RAM. Use Project/Exit Editor to exit the editor completely.

REXX DEBUGGER

Activates debugger information output while processing incoming Rexx commands: Commands arriving at the Rexx port are displayed in the shell window GoldED has been started from (you will see no debugging output if GoldED has been started via the Workbench).

DISABLE FONT CACHE

Deactivates the CPU caches while generating fonts. On some machines random pixels appear in text windows if using very small fonts (width below eight pixels, e.g. xen font, thinpaz font). This is probably a bug in the operating system. Enable this option and restart GoldED to see if it fixes the problem. Don't use fonts smaller than eight pixels (Configuration/Display) if you are still experiencing display problems.

NUMPAD = MOVEMENT

Enables or disables the alternate configuration of the numerical keypad.

UNDO RAM WARNINGS

The editor usually warns you if the backup data to be handled by the undo feature doesn't fit into the undo buffer (you will then not be able to undo the last operation). Increase the undo buffer size if these warnings appear frequently. No warnings are displayed if you uncheck this checkbox.

REDUCED CHIP-RAM USAGE

Selects 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 open windows: this option sets the refresh strategy for windows you open in the future. Two strategies are available: window refresh performed by the operating system and window refresh performed by GoldED.

- Refresh performed by the OS (REDUCED CHIP-RAM USAGE disabled)

  The operating system will refresh the windows by using CHIP RAM to buffer hidden parts. Advantage: fast. Disadvantages: high CHIP RAM consumption. Performance suffers drastically if you have many overlapping windows which all require buffering.

- 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 (overlapping) windows. Disadvantage: Delayed refresh if GoldED is busy.

UAE users should always use GoldED's built-in refresh ("reduced chip ram usage") because the memory bandwidth of the emulator is limited so that using buffers for the display refresh makes little sense.

CONFIRM DUPLICATES

Enable CONFIRM DUPLICATES to permit loading the same file multiple times (you will be notified if a file has already been loaded). GoldED immediately activates the open file if duplicates are not permitted.

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 while position and size of the Reqtools requester are set by the Reqtools preferences program.

BRIEF MESSAGES

Enables brief messages in the status bar (as opposed to error requesters which have an "ok" button and require your confirmation).

TRANSPARENT SYMBOLS

Pen 0 in symbols is considered to be the background pen and rendered transparently if this option is checked (to avoid a visible border around icons if the screen’s background color doesn’t match the icon’s background color).

CACHE FOR IMAGES

Images accessed by GoldED are saved to the main configuration file (golded:etc/registry/registry) if this option is enabled. While this feature does increase RAM consumption (the configuration file is loaded into memory every time you start GoldED), it can significantly reduce the startup time of GoldED by reducing the number of files to be searched on the disk.

FAST SEQUENCES (SEQUENCE COMPILER)

Recorded sequences are compiled into a fast format directly after recording if this option is enabled. For example, multiple keystrokes are converted to a single TEXT command. Sequence compilation can introduce subtle changes into the macro. For example, multiple keystrokes do not always produce the same result as a single text command (especially if you have enabled features like automatic case correction or templates).

CONFIRM BACKUPS

Automatic backups are not performed until you have confirmed the backup if this option is enabled.

CONFIRM ERRORS

Error requesters block the editor and require your immediate confirmation if this option is enabled.

CORRECT FILE NAMES

Shows if minor spelling errors in file names are corrected automatically (for exmaple, ’main.c’ is loaded when you accidentally type "nain.c").

UNFOLD TREEVIEWS

Shows if the first level of treeviews is unfolded automatically.

Extras/Customize/Global Options/Misc

BACKUP INTERVAL

AutoBackup

Sets the interval for automatic backups. Units are minutes (0 deactivates backups). Backups are saved to a configurable backup directory (Configuration/Misc) which may be the same directory the document is saved to. GoldED adds "copy of" to the names of backup files. Enable the "Confirm backups" option (see above) if you want to confirm automatic backups.

PASSWORD

Global password. Used when saving (or reading) encrypted XPK files (see Project/Save Compressed File).

DATE FORMAT

Format for date strings. Changes of the date format don’t take effect until you restart the editor. The following placeholders can be used in the date string:

```
%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

The format of window titles. The first "%s" found in this template is replaced by the file name, the second "%s" is replaced by the filetype (Filetypes) of the document.

1.111 Extras/Customize/Filetypes

Extras/Customize/Filetypes of:
EXTRAS MENU
Basis of GoldED's configuration system are presets and ↩
filetypes: presets
(menu configuration, keyboard configuration, display configuration, etc.) are assigned to a document when the document is loaded. Presets are selected from a large pool of available presets based on the filetype assigned to the document (which is determined by looking at the file's name). The presets currently assigned to a file appear in the context menu of every text window as "Filetype Settings".

Filetypes

The filetype tree shows the user-defined filetypes. Each filetype has a unique name (e.g. "SAS/C") which you can choose freely, an associated list of presets (its properties) and an associated list of file name patterns (e.g. "#?.c"). The patterns controls if and when the filetype is used. For example, the "SAS/C" filetype is used for C source code files which have file names of the form "<filename>.c" (e.g. "main.c"). This means that the presets associated with the SAS/C filetype - its properties - are used when you load a C source code: the editor will use the menu chosen for C source codes, the display configuration chosen for C source codes, etc.

Filetypes are priority-sorted: The list is scanned from top to bottom when the editor is looking for an appropriate filetype for a new text. The last filetype is the "default filetype" because it is used if no suitable match is found. The name of this filetype should be "TEXT". You can determine the filetype chosen for a document by looking at the title bar of the window: the filetype name is displayed after the file name.

Some filetypes don't seem to have an associated AmigaDOS-style file name pattern. For example, the "Joyce" filetype (shipped as add-on with GoldED) has the string "JOYCE" in its pattern list which obviously isn't a pattern. These filetypes must be chosen manually if you want to use them: open the document's properties window (
Project/Properties
) and enter JOYCE into the
filetype field. Or use the
SET
command (example: SET TYPE="JOYCE").

Note: Internally, GoldED uses expanded file names. All symbolic assigns are expanded to the real path of the file. For example, the internal name of the file "s:user-startup" could be "dh0:s/user-startup". File name expansion must be considered when choosing filetype patterns.

Hierarchical configuration

The list of presets associated with a filetype may be incomplete: GoldED uses the standard presets (e.g. the "standard.menu") if the filetype does not specifically require another preset. Activate the properties list of a

filetype and click on the "+" to add a preset to the filetype. Select a preset and click on the "x" gadget to delete it from the filetype's property list (the preset file is not actually deleted). Doubleclick on a preset to modify it (
Configuration
).

Pool

Click on the "Pool" button to see a list of available presets. The presets are grouped into categories ("Menu", "Mouse", "Display", etc.) to simplify navigation. You can add new presets or delete presets. The only way to add a new preset to the pool is to duplicate one of the existing presets. You should be careful when deleting a preset from the pool list because the preset file is actually deleted from the disk (standard presets can not be deleted).

Technical information:

Preset files are stored in the drawer "golded:etc/registry/presets/". The filetypes setup is saved to the file "golded:etc/registry/registry". These files and directories may not be modified manually. Do not copy files to these directories and do not delete or rename files. Ignoring this warning could destroy the configuration. Instead, use the "regedit" program shipped with GoldED (golded:etc/add-ons/regedit) to install preset files. Use GoldED's pool requester to delete files.

1.112 Configuration

Configuration

Doubleclick on a preset listed in the
Filetypes
tree to configure a preset.
Or choose the preset to be configured from the context menu of a text window. The advantage of using the context menu is that you are guided directly to the presets attached to the text. The following preset types are available in GoldED:

Configuration/Context

Configuration/Plug-Ins

Configuration/Display

Configuration/Printer

Configuration/Dictionary

Configuration/Project

Configuration/File Search

Configuration/Reference Files

Configuration/Keyboard

Configuration/Syntax Highlighting

Configuration/Layout

Configuration/Tabs

Configuration/Menus

Configuration/Templates

Configuration/Misc

Configuration/Toolbars

Configuration/Mouse

Configuration/User Interface
Hierarchical Configuration

The presets used by GoldED are based on preset files which are stored in the "golded:etc/registry/presets" drawer. For example, the standard menu is saved to the file "golded:etc/registry/presets/standard.menu". While this file-oriented design doesn't seem to support hierarchical configuration, hierarchical configuration actually is available: many configuration dialog boxes (menu, keyboard, toolbars) have an "include" button which can be used to include the configuration of other filetypes. For example, you can create a keyboard preset for writing e-mails which includes the keyboard configuration of the filetype "TEXT" and redefines a few keys only.

1.113 Configuration/Context

Configuration/Context of the
Configuration
menu

Dialog box for setting the context-sensitive indention scheme. The settings in this dialog control the cursor position after a linefeed.

SmartIndention

The indention scheme (ie. the new cursor position) depends on keywords in the cursor line: the first or the last word in the cursor line determine the indention scheme to be used. For example, you can instruct the editor to indent the cursor after a line containing an "if" statement: Add "if" to the list of keywords and set the indention for this keyword to "cursor right".

1.114 Configuration/Display

Configuration/Display of the Configuration menu

Display configuration dialog: sets the screen resolution, the fonts etc. You shouldn't modify the display configuration if running out of (CHIP-)RAM because the editor might not be able to allocate enough RAM to reopen the screen.

Configuration/Display/Screen
Configuration/Display/Colors
Configuration/Display/Fonts
Configuration/Display/Windows
Configuration/Display/Details

Configuration/Display/Screen

Shows the screen mode to be used for GoldED. We recommend to use GoldED on its own screen (if possible) because some options from the editor's display configuration will have no effect if GoldED is forced to run on a public screen (because these options are set by the owner of the screen).

The recommended display mode for GoldED is 800x600 pixel and 256 colors. This display mode is only available if you have installed a graphics card in your computer. GoldED can not be used with resolutions below 640x400 pixel. Symbols used in the program have 16 colors but the screen mode used with GoldED should have at least 32 colors because GoldED occasionally needs more free colors (and other applications running on the editor screen need a few free colors, too).

The recommended display mode for UAE users is "uaegfx:800x600 16bit PC" with the actual screen depth set to 256 colors (using less colors makes the editor run slower).

Configuration/Display/Colors

PENS

GoldED supports AmigaOS 3 compliant pen management: You specify the colors you want to see used for rendering user interface elements (the "Pens" list) and GoldED will attempt to allocate these pens from the OS. Click on the "+" button to add a new pen. Click on the "view" button to set its RGB value. There is no limit to the number of pens you define. However, the number of pens actually made available to GoldED by the OS depends on the screen mode: on a 16 colors screen, 16 colors can be allocated in the best case (less than 16 colors if some pens have already been allocated by other applications or the OS itself). The editor will automatically use the closest match from the pool of available pens if a pen allocation fails. You should have the important colors (black, white, etc.) at the top of the pen list since these pens are allocated first and thus have the best chance of a successful allocation.

Colors samples

The display representation of the RGB values in this dialog could be incorrect if the current display mode does not provide enough free pens to show correct color samples. On true color cybergraphics screens the list of pens will always show correct color samples but the number of pens actually made available to GoldED at run-time still depends on the number of free pens.

USER INTERFACE

The list of configurable user interface elements. Select an element and doubleclick on a pen (in the pen list) to set its color. Some user interface elements can only be configured if GoldED is running on its own screen. For example, the window border color and the menu color can not be set if GoldED is running on the Workbench screen because these colors are set by the screen owner.

Configuration/Display/Fonts

Here you can choose the fonts to be used for the text, for the preview display, for requesters and for menus (if the editor is running on a custom screen). The text font and the preview font must be fixed width fonts. All other fonts should be proportional fonts. Adjust the requester scaling (Configuration/Display/Details) if gadget labels appear to be too large for some gadgets (ie. if you see text overwriting gadget borders).

Rendering bug

Some Amigas have problems with fonts smaller than 8 pixels: Random pixels appear in the text. This problem can not be fixed as it seems to be a bug of the OS (it is blitter-related and does not affect Amigas with a graphics card): you will have to use fonts wider than 7 pixels if your Amiga is affected. Specifically avoid the XEN fonts and the thinpaz fonts which are both seven pixels wide.

Configuration/Display/Windows

REQUESTER SIZE (REQUESTER SCALING)

Configures the requester dimensions. "0" means that all requesters are scaled to match the font you are using. Positive values enlarge the requesters. You can correct the aspect ratio of dialog boxes by setting the scaling factors for width and height to different values.

BUTTON SIZE

The minimum size for toolbar buttons. The actual toolbar button size depends on the toolbar images, too.

SLIDER SIZE

These gadgets set the width and height of sliders in window borders.

BORDERS

These input fields define the screen border area to be kept free when arranging windows.

Configuration/Display/Details

CHUNKY PIXEL HARDWARE

External graphics boards - unlike the original Amiga graphics hardware - do not use bitplane-oriented hardware for screen modes with 256 or more colors. Instead, they use a chunky pixel organization for their display memory. Usually GoldED will try to improve rendering performance by restricting output to single bitplanes. While this strategy does increase performance for native Amiga chipsets, it might have no effect when using modern graphics hardware (or even decrease performance). That's why bitplane masking can be disabled with this checkbox. Note that even chunky pixel hardware can profit from bitplane masking. For example, UAE with Picasso96 drivers usually is faster with the chunky-pixel option disabled even when using chunky pixel true color modes. You should always try both settings to find out which is faster.

IGNORE OVERSCAN

Functions related to arranging windows (
View/Windows/Arrange Vertically
)
usually determine the screen's visible display rectangle and try to arrange the windows within this area. Enable the ignore-overscan option to have the screen's real size considered instead.

CENTER WINDOWS

Activate this option to have new windows opened centered on the screen.

ARRANGE WINDOWS

Shows if windows are rearranged after a window has been closed (or a new window opened).

PRIORITY FOR FOCUS WINDOW

Sets the weight for the current window when arranging windows. Example: Set the weight to 2 to make the the current window twice as large as the other windows.

SHOW BOTTOM SLIDER

The bottom scrollbar can be turned on or off.

SHOW SYMBOLS

Symbols in GoldED's dialogs, lists and treeviews can be turned off to increase performance on slow computers.

SHOW TIME AND DATE

Shows if time and date are displayed in (before) the screen's title bar.

BLANK BORDERS

Shows if screen borders should be black. This option has no effect if border blanking is not supported by the graphics hardware of your computer.

1.115 Configuration/Dictionary

Configuration/Dictionary of the
Configuration
menu

Dictionary configuration. Several input aids provided by GoldED depend on this dictionary, including AutoCase (automatic case correction) and expression completion (
APC
).

AutoCase

Case of words you add to a text is checked if this option is enabled. Checks are performed when the cursor leaves a modified line.

Tips

Don’t let the dictionary grow too big. A large dictionary reduces usability of the
APC
function because 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".

1.116 Configuration/File Search

Configuration/File Search of the
Configuration
menu

This dialog box lists the places to be searched by the
Search/Find File
function. Subdirectories are examined if the "Recursive" option is ↩
enabled.

1.117 Configuration/Keyboard

Configuration/Keyboard of the
Configuration
menu

Keyboard configuration. The keyboard requester is used to configure keys (and key/qualifier combinations). GoldED recognizes the qualifiers ALT, CTRL and SHIFT or a combination of these keys (AMIGA keys are reserved for the menus). The system's default keymap is used for unconfigured keys: The "A"-key inserts "A" into the text if unconfigured. Control keys (e.g. cursor keys) are initially undefined. GoldED is shipped with all control keys configured to useful defaults. Configuration of a key(-combination) is easy: Use the "record" button to add a key to the list of configured keys. Doubleclick at any of the listed keys to modify it. Key configuration is similar to menu configuration (
Event Definition
).

Dead keys

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

1.118 Configuration/Layout

Configuration/Layout of the
Configuration
menu

Layout configuration. Layout borders (left margin, right margin) set by this requester are related to formatting functions only, they do not restrict cursor movements. 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 (the word parts are concatenated again).

1.119 Configuration/Menus

Configuration/Menus of the
Configuration
menu

Menu configuration

Menu titles, menu items and subitems are displayed in the menu tree and can be modified. The number of menus, items and subitems is limited by the AmigaOS. For example, you can not have more than 32 items in each menu. Doubleclick on an item to configure it (Event Definition). Item configuration is similar to keyboard configuration, mouse configuration, templates configuration, etc.

File list

Add the file list placeholder to a menu if you want to have all open documents (hidden buffers included) listed in same menu. This feature affects performance on slows Amigas because the OS functions used to maintain the menu are slow.

1.120 Event definition

Event definition

The event definition requester is used to configure menu items, keys, mouse buttons and various other resources of GoldED. It allows you to assign internal commands, macros, text, etc. to these resources: Simply add the command, scripts, text etc. to the event list in this dialog box.

Magic codes

Event list entries of the type DOS, REXX or TEXT are "interpreted": Reserved keywords found in these entries are replaced by a corresponding value at execution time (unless the keywords are quoted). The following keywords and syntax elements are recognized:

```
"text" .............. Simple text
\" .................. 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)
```

Note: Rexx macros are started asynchronously. Event processing will not stop. Event configuration should therefore not depend on assumptions about if or when a macro has been processed.

OUTPUT

Sets the output path or the output device. Magic Codes can be used in this text field. Example: con:0/0/640/400. The default console is used as output device if this field is left empty (Configuration/Misc).

SHANGHAI

The shanghai interval. 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.

HOTKEY

Menu shortcut definition. This field is ignored for non-menu events. Click at the arrow button to see a list of unused shortcuts. Text you enter into this field can not be used as menu shortcut if you enter more than one character: Instead, the text appears in the menu literally.

VARIABLE

Variable name. This field is ignored for non-menu events. A checkmark is added to the menu if you specify a variable name in this field. The (boolean) variable value is used to control the checkmark state. Note that the menu won't change the variable state when used: attaching a variable to a menu is a rendering operation only. You'll have to add appropriate commands to the menu (see SET command) if the menu should control the state of the variable.

GUIDE

This field lists a help text or a guide page (from an AmigaGuide document) descibing the event. Online help for menus is displayed if the user holds down the HELP key while selecting a menu (MenuHelp). Online help for toolbar buttons is displayed if the user moves the mouse pointer over a toolbar gadget (and waits for a second). Enter the name of a guide file followed by "@" followed by the name of the guide page. You may leave out the name of the guide file and the "@": GoldED will then use the guide file of the menu (Configuration/Menus). Short help texts for toolbar buttons can be specified directly: enter the help text (put into quotation marks) directly into this field.

1.121 Configuration/Misc

Configuration/Misc of the Configuration menu

This dialog lists miscellaneous configuration options:

Configuration/Misc/Options
Configuration/Misc/Files

Configuration/Misc/Fold
Configuration/Misc/Backup

Configuration/Misc/Options
SAVE DOCUMENTS WITH ICONS

Shows if icons are created for documents saved by GoldED.

SAVE BREAKPOINTS

Shows if breakpoints are saved with the document (so that breakpoint states can be restored quickly to the last 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 with breakpoints (Debugger) and they are loaded automatically if available.

FILE PROTECTION BITS: READABLE, WRITABLE, EXECUTABLE, DELETABLE, SCRIPT

These checkboxes determine the default bits for a new text created by View/Open New Window . Use Project/Properties to modify protection bits of open buffers.

OVERWRITE WITHOUT WARNING

Shows if existing copies of a document are overwritten without warning.

Configuration/Misc/Files

DEFAULT TOOL

Default tool of icons created by GoldED.

XPK MODE, XPK EFFICIENCY

Click on the arrow button to see a list of available XPK compressors (stored in the libs:compressors directory). The compressor you select is used for saving XPK files (Project/Save compressed file). Additionally you can choose the efficiency for compression (0% to 100%). Please read the original XPK documentation for an introduction into the XPK software. Some XPK compressors require a password (Extras/Customize/Global Options).

CONSOLE

Standard output device used when running external programs and macros. The console description is interpreted, ie. you may use the Magic Codes

in
the console description.

INFO FILE

The icon file used for saving documents with icons.

Configuration/Misc/Fold

FOLD START, FOLD END

Fold markers (see
Folding
). Start marker and end marker must be different
if you plan to use nested folding (folds within folds).

FOLD ALL FOLDS WHEN LOADING

Shows if files containing fold markers are folded automatically after loading.

Configuration/Misc/Backup

You can configure the editor to create a backup of the previous version when overwriting a document. Both, the backup path and the number of backups are configurable.

BACKUP PATH

Backups are saved to the backup path. You can leave this field empty if you want to have backups saved to the document’s drawer. The backup directory should be on the same disk/partition you are usually using for storing documents because the editor can then create backups by renaming the old file (as opposed to copying it which is much slower).

GENERATIONS

Sets the number of backup generations per file. For example, set this to five tho have the five last versions of a file available in the backup path. Consider the amount of disk space used up by this function when changing this setting.

The editor will choose filenames of the form "ID.Name.Generation.Suffix" for backups. For example, the two first backup generations of the file "main.c" will be named "xxx.main.1.c" and "xxx.main.2.c". The ID part is meant to stop backups of files with the same name (but a different path, e.g. "dh0:main.c" and "dh1:main.c") from overwriting each other.

BACKLOG [days]

It is often desirable to find older versions of a document in the backup path. For example, if you are saving a file "main.c" daily, you probably do not only expect to find a backup of the version from yesterday but you also whish to find the version from last week. Setting the generations slider to a high value could solve this problem but only at the expense of using up much disk space. The backlog slider offers you a compromise between disk usage and backups that are reaching far back in time: the editor will continue to create only a small number of backup generations but spread these files over the period you have specified. For example, set the backlog to 21 days and the generations slider to five generations. When saving daily, you will find five backups of each file spread over the last three weeks.

Note: If the BACKLOG setting is numerically smaller than the GENERATIONS setting (for example three generations, four days backlog), then the editor will create only one backup generation for the current day. The other backup generations are reserved for the previous days. You will have to set BACKLOG to a smaller value than GENERATIONS if you want to have more than one backup for the current day.

1.122 Configuration/Mouse

Configuration/Mouse of the
Configuration
menu

Use this requester to configure the mouse buttons and the context menus.

Mouse configuration

The left mouse button and the middle mouse button - if available - can be configured. The right mouse button is reserved for accessing the menu. Single clicks, double clicks and combinations with CTRL, ALT and SHIFT can be configured. Some combinations are not be available for configuration because they are consumed by the OS or other utilities. For example, CTRL plus a mouse button is used by the OS for screen dragging. Event definition is similar to menu/keyboard configuration. See
Event Definition
for
details.

Context menus

Click with the right mouse button over a text window to open a context menu. Context menus pop up under the mouse cursor and list context-specific commands. For example, the context menu shows block-related commands if text has been marked. GoldED defaults to using the right mouse button for context menus which is the standard invocation method used by various other operating systems. On the Amiga, using the right mouse button does conflict with using the same button for accessing the normal menus. GoldED attempts to solve this conflict by dynamically using that button either for showing the context menu or for accessing the normal menu based on the position of the mouse pointer: Right-click over a GoldED window and you will see the context menu. Right-click over the menu bar (or simply outside a GoldED window) and you will see the standard menu. This mechanism can be turned on/off with the "Right mouse shows context menu" switch.

The configuration of context menus is similar to the configuration of menus
(
Event Definition
): Doubleclick on a menu item to open the event

configuration dialog. The name of a context menu (the "context condition") controls if and when the items of the menu appear in the context menu at invocation time. The syntax for context conditions is:

<variable> <operator> <argument>

Supported operators for context conditions are "=" (equal) and "!=" (not equal). <variable> can be any of GoldED's built-in variables (
Variables
)
or a user-defined variable (
Extras/Customize/Global Options
). Example:

POSITION=TEXT

This context condition means that the associated items will appear in the context menu if the variable "POSITION" has the value "TEXT" (at invocation time, ie. when the user is pressing the right mouse button to see the context menu). The variable POSITION is one of GoldED's predefined variables (see
Variables
): it describes the mouse pointer position.
POSITION will have the value TEXT if the mouse pointer is over the text area of an editor window.

Tip: You can use the empty condition "*" (which is always "true") to unconditionally add items to the context menu.

1.123 Configuration/Plug-Ins

Configuration/Plug-Ins of the
Configuration
menu

API interface configuration (API stands for "Application Programmer Interface"): Add the plug-ins you want to use to the list of clients. Clients are editor extensions adding additional features to GoldED. Some clients require startup options: doubleclick on a client to set its startup arguments.

Technical information

API clients are libraries. You may not rename clients because the AmigaOS doesn't permit renaming libraries (renamed API clients can not be started). Be careful when adding API clients to the list to specify the name correctly (the name is case-sensitive). Specifications, example source codes, API clients, includes etc. can be found in the golded:developer/api drawer.

1.124 Configuration/Printer

Configuration/Printer of the
Configuration
menu

GoldED uses the Workbench preferences printer. However, a few printer-related options (linefeed, spacing, etc.) can be set directly in GoldED, too.

INIT

The content of this string button is sent to the printer after basic printer initialization. The init string can be used to pass additional control codes to the printer device. The init string is interpreted (see

Magic Codes
) so that non-ASCII codes can be sent to the printer. Normally, the printer device doesn't allow you to pass modell-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 button to have data sent to the printer in the background: Printing won't block text input and 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 do not affect the output.

1.125 Configuration/Project

Configuration/Project of the
Configuration
menu

Project management: Add the 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 itself except that you can run a user-defined make macro by using the Make button in the project requester (the selected file tree is passed to the macro).

Technical information for programmers

External programs can read the project list by sending the
QUERY
command to

GoldED’s Rexx port (QUERY PRJLIST). The editor returns a pointer to a read-only object list. This pointer stays valid between

LOCK

and

UNLOCK

and only until the next Rexx command is sent to GoldED. Example ↩

source code

for reading the list is available in the developer drawer (golded:developer/examples/project).

1.126 Configuration/Reference Files

Configuration/Reference Files of the

Configuration

menu

Configuration requester for the reference system. The reference system is based on databases: databases are collections of keywords. For each keyword, a path to a file containing keyword-related information is stored in the database. Databases are built automatically by GoldED. All you have to do is to add files and directories containing information to the list and to press the "Generate" button. Database creation will take a long time (minutes) if many files are to be scanned. GoldED will scan the files, extract keywords and save the keywords to the database file(s). Various file formats are understood by GoldED: Keyword extraction is performed by parsers made for parsing specific file types. For example, C sources are parsed by the C parser. 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. External parsers can be added in order to increase the editor’s capability to understand more file types. Parsers are added and configured using

Search/Show Function List

menu.

1.127 Configuration/Syntax Highlighting

Configuration/Syntax Highlighting of the

Configuration

menu

Syntax Highlighting

Syntax highlighting preferences

GoldED can display color coded text to increase readability. The 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 and 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 code is included for those who want to write their own syntax parser (golded:developer/examples/syntax).

Performance of syntax highlighting very much depends on the graphics hardware of your computer, not on the CPU power: a 68030/ECS Amiga is too slow for syntax highlighting while the same Amiga with a PicassoIV board has no problems with syntax highlighting in 256 colors.

PREPARSE DOCUMENT

Syntax parsing is performed either immediately after loading a document (preparsing enabled) or on-demand when the document is displayed: on-demand parsing reduces the load time (and initial 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. Preparsing should be enabled for fast parsers only.

PARSER

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.

1.128 Configuration/Tabs

Configuration/Tabs of the
Configuration
menu

Tabulators

This is the tab configuration dialog. GoldED supports various tab modes: regular tabs (spaced at a constant distance), tabs at user-defined positions, tabs which are expanded to spaces, etc. Use the TAB key (->|) to insert tabs or the TAB key plus SHIFT for the BACKTAB function (|<-).

Tab key function set to "Move Cursor"

The recommended setting for GoldED is to configure the TAB/BACKTAB key to move the cursor to the next/previous tabstop and to use spaces (not TAB codes) to indent lines: Real tabs provide no advantages for the simple task of indenting lines compared to the powerful indention functions available in GoldED. Besides, files without tab codes can be loaded faster and are compatible with all other editors and printers (which is not necessarily true for files with real tabs). The sole advantage of using real tabs is that the final file size will be slightly smaller but the size of text files shouldn’t be an issue these days.

Real Tabs

Set the button to "Insert Tabs" if you prefer to use real tabs (as opposed to have the cursor moved or spaces inserted when using the TAB key). You're probably already used to tabs in word processors: they provide ways and means to align text set in a proportional font, they help you to create tables, etc. Tabs in a text editor are different because the ASCII format does not permit saving tab stop positions. That's why all editors have to make assumptions about the position of tab stops and what tab codes in files mean. They usually assume that tab stops are set every <n>th column. <n> traditionally (since the days of the first printers) is 8 but GoldED can be configured to any tab size between 1 and 8 with the "Tab size in files" gadget. Today, most programmers prefer to use a much smaller tab size on the screen (usually 4 columns). GoldED supports these programmers by permitting different tab stops for open buffers ("Tab stop positions" gadget), ie. while the file is loaded by GoldED. However, when saving the file, the tab scheme is converted back to the standard tab size of the operating system so that files saved by GoldED can be viewed with other programs. This conversion might require conversion of some or all tabs to spaces. No conversion to spaces will take place if the tab size on screen and the tab size in files are identical (e.g. both set to 4 which is the recommended setting). Or if the tab size on screen is an even multiple of the tab size in files. All other configurations possibly require some conversions.

Technical information

Most editors provide tab support based on the concept of a glyph of variable width: Internally, tabs are stored as ASCII code 9. This code is expanded to the appropriate number of spaces when the line is displayed. This concept has two disadvantages: scrolling speed is affected by TAB expansion and the glyph concept requires the introduction of an additional system of coordinates. GoldED uses a different approach: TAB codes are expanded after the text is loaded and converted back to TAB codes before the text is saved. TAB expansion is moved from the display routine to the load/save functions as far as GoldED is concerned. The advantage of this approach is that tabs don't come with a performance hit. The disadvantage is that tabstops can not be changed dynamically (ie. changing tab stop positions does not affect the indention scheme of open buffers).

User-defined tabstops

Arbitrary tabstop positions should not be combined with real tabs because the ASCII file format does not permit saving arbitrary tab stop positions. Instead, you should configure the TAB key to "Move cursor" or "Insert spaces" before using user-defined tabstops.

1.129 Configuration/Templates

Configuration/Templates of the
Configuration
menu

This is the template configuration dialog (
Templates

). Templates are patterns the editor is looking for when you enter text (provided that templates detection has been enabled:
Layout/Detect Templates
). The action assigned to a template is performed when it is detected in the input stream. Only single words (no spaces) may be added as templates. Doubleclick on one of the templates displayed in the list to configure the action. Template configuration is similar to configuration of menu items (
Event Definition
).

1.130 Configuration/Toolbars

Configuration/Toolbars of the
Configuration
menu

GoldED supports two types of user-defined buttons: text buttons in window borders and toolbar buttons with images. Add the images (in IFF format) to the toolbar tree and the text labels to the list of window border buttons. Doubleclick at a button definition to modify the action associated with the button (
Event Definition
). Toolbar images designed for GoldED traditionally have a one pixel wide border which is removed automatically before the image is displayed in a toolbar.

Colors

The colors used for rendering the images are not the true colors of the image but the pens defined in the display configuration of GoldED (
Configuration/Display/Colors
). That’s why the palette of your own images (if you plan to add any) should match the pens defined in the display configuration. The best way to ensure this is to load one of the images shipped with GoldED into your paint program and to use its palette.

1.131 Configuration/User Interface

Configuration/User Interface of the
Configuration
menu

This dialog set various user interface options:

Editing

ACCELERATE SCROLLING AT END OF PAGE

Shows if scrolling speed is increased when the cursor reaches the end of the page.

STOP CURSOR AT END OF LINE

Enable this option to have the cursor moved to the beginning of the next line when it passes the last character of a line while CURSOR-RIGHT is pressed.

CURSOR DOESN'T BELONG TO BLOCK

Determines whether the cursor is part of the block while marking. You might have to adjust the cursor color (Configuration/Display) after changing this option in order to visually separate block and cursor.

CHECK BRACKET NESTING WHILE TYPING

Shows if the editor automatically checks the correct use of round brackets "()" in the current line. This check is performed when the cursor leaves a modified line.

KEEP INDENTION

Shows if the indention is kept after a linefeed.

DRAG & DROP FOR BLOCKS

Shows if marked text can be moved or copied with the mouse. Click at marked text and hold the mouse key down for a moment. This will put the editor into drag and drop mode (the mouse pointer changes from an arrow to a rectangle). You can now move the marked text to the new position by releasing the mouse button over the destination point. Press the STRG key while in drag & drop mode to copy the text (as opposed to moving it).

INPUT REPLACES BLOCK

Shows if input replaces the marked text if the cursor is positioned over marked text (this option does not affect line blocks and vertical blocks).

Interface

SCROLL BORDERS

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

WHITE SPACE

Defines what characters should be handled as if they were spaces ("white space characters"). This definition is evaluated by many functions. For example, including "(" to the list would make the 'jump to next word' function (SHIFT CURSOR RIGHT) consider the "(" as word separator.

```
Syntax of the white space definition string: a list of  ASCII  codes  (e.g.
128),  ASCII code ranges (e.g 128-160) and strings in quotation marks (e.g.
" "). All elements must be separated by colons, not by spaces. Example:

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

1.132 Extras/Save Settings

```
Extras/Save Settings of:
 EXTRAS MENU
Saves the current configuration.
```

1.133 Macro Menu

```
                                Macro Menu
            Menu tree of macro menu



  Macros/Load Session

  Macros/Sequence Play Loops

  Macros/Save Session

  Macros/Sequence Apply To

  Macros/Start Text As Macro

  Macros/Load Sequence

  Macros/Run Macro

  Macros/Save Sequence

  Macros/Edit Macro

  Macros/Save As Rexx Macro

  Macros/Macros

  Macros/Repeat Input

  Macros/Sequence Record

  Macros/Execute Command
```

Macros/Sequence Play

1.134 Macros/Load Session

Macros/Load Session of:
MACRO MENU
Loads a session file saved by
Macros/Save Session
and restores the
environment of that session: 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.

1.135 Macros/Save Session

Macros/Save Session of:
MACRO MENU
A description of the current working environment is saved to a ↩
session file
(position and size of text windows and a list of the hidden documents are saved to the session file). All modified text buffers are saved if requested. Use
Macros/Load Session
to restore the environment described by a session
file.

1.136 Macros/Start Text As Macro

Macros/Start Text As Macro of:
MACRO MENU
Executes the current text as Rexx macro. This manual doesn’t ↩
contain a
description of Rexx nor an introduction into Rexx programming. Please contact your local bookstore for a book on Rexx.

You should save the text before you call this function because the version on disk is executed, not the text in memory. All Rexx macros must start with a comment (/* ... */) so that the Rexx server does recognize the text as macro. Nothing will happen if the text does not start with a comment. A macro executed by this functions has its host (GoldED) set up automatically, i.e. you don’t need an ADDRESS command in the macro. However, you will have to select a host manually with the ADDRESS command if the macro is started by the AmigaDOS command RX and not by GoldED. The example below shows how a macro can check if it has been started by the editor:

```
/* $VER: 1.0, ©1996 Dietmar Eilert. Empty GoldED macro */
```

```
OPTIONS RESULTS                             /* enable return codes     */

if (LEFT(ADDRESS(), 6) ~= "GOLDED") then    /* not started by GoldED ? */
    address 'GOLDED.1'

'LOCK CURRENT RELEASE=4'                    /* lock GUI, gain access   */
if (RC ~= 0) then
    exit

OPTIONS FAILAT 6                            /* ignore warnings         */
SIGNAL ON SYNTAX                            /* ensure clean exit       */

/* ------------------------ 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 ":" ERRORTEXT(RC) ":-("
'UNLOCK'
EXIT
```

1.137 Macros/Run Macro

Macros/Run Macro of:
MACRO MENU
Executes a macro file. Rexx macros can be found in the "golded:etc/rexx" drawer. You will rarely need this function because macros usually are attached directly to menus or keys (see
Event Definition
)

1.138 Macros/Edit Macro

Macros/Edit Macro of:
MACRO MENU
Loads a macro file from GoldED's Rexx directory (golded:etc/rexx). Rexx macros should have the file name suffix ".rexx" (e.g. number.rexx). All macros designed for GoldED must follow a special protocol to register with GoldED before performing any operations to prevent race conditions with the

user or other macros (see Rexx Port). You should use the empty macro "empty.rexx" shipped with GoldED as basis for developing your own macros.

1.139 Macros/Macros

Macros/Macros of:
MACRO MENU
This menu lists the Rexx macros shipped with GoldED:

Macro Number Lines
Macro Insert File List
Macro List Directory
Macro Fold Block

MACRO NUMBER LINES

Numbers all lines of the document. You choose the first line number and the interval for line numbers.

MACRO INSERT FILE LIST

Inserts a list of file names (chosen from a file requester) into the document. Use SHIFT to select multiple file names in the file requester.

MACRO LIST DIRECTORY

Lists the contents of a directory.

MACRO FOLD BLOCK

Folds the marked text (see Block/Mark Text on how to mark lines). Read the Folding section of this manual if you are unfamiliar with GoldED’s folding feature.

1.140 Macros/Sequence Record

Macros/Sequence Record of:
MACRO MENU
Sequences

Starts (first call) or stops (second call) recording of a sequence. The shortcut for this command is SHIFT-F10. During recording all keystrokes and all menu selections are recorded. Mouse movements and mouse clicks are not recorded. Use Macros/Sequence Play to replay a recorded sequence. Use

Macros/Save Sequence to save the sequence or Macros/Load Sequence to load a sequence. Saved sequences can be assigned to keys, toolbar buttons, menus, etc. (see Event Definition).

1.141 Macros/Sequence Play

Macros/Sequence Play of:
MACRO MENU
Replays a recorded sequence (see Macros/Sequence Record). The keyboard shortcut for this function is F-10.

1.142 Macros/Sequence Play Loops

Macros/Sequence Play Loops of:
MACRO MENU
Replays a 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).

1.143 Macros/Sequence Apply To

Macros/Sequence Apply To of:
MACRO MENU
Applies a sequence to a list of files. Record a maco, then use this function.
GoldED will load the files you choose in the file requester (SHIFT-click to select multiple files) and run the sequence for each of the files.

1.144 Macros/Load Sequence

Macros/Load Sequence of:
MACRO MENU
Loads a sequence from disk.

1.145 Macros/Save Sequence

Macros/Save Sequence of:
MACRO MENU
Saves a recorded sequence (Macros/Sequence Record). Sequences should be saved to golded:etc/recordings and the file extension should be "*.seq".

1.146 Macros/Save As Rexx Macro

Macros/Save As Rexx Macro of:
MACRO MENU
Saves a recorded sequence (Macros/Sequence Record) as Rexx macro. Rexx macros are text files and can be edited with GoldED (Macros/Edit Macro).

1.147 Macros/Repeat Input

Macros/Repeat Input of:
MACRO MENU
This dialog box sets the repeat count fore the next event. Example : enter a repeat count of 80 and press the "-" key. This will insert a horizontal line.

1.148 Macros/Execute Command

Macros/Execute Command

Shows the command dialog. Enter the internal command you want to have executed (see Internal Commands). The shortcut for this function is SHIFT-ESC. Example: INFO VERSION

1.149 About GoldED

About GoldED

Shows general information about GoldED (program version, the name of GoldED's Rexx port and the current screen name). Editor screens are public, i.e. you can run other programs on the same screen. The following example opens a shell window on the editor screen (note that there is no space between the "screen" keyword and the screen name) :

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

1.150 Keyboard

Keyboard

Cursor keys

TAB key

HELP key

ESC key

RETURN key

F-keys

DEL key

SPACE key

Some very useful functions of GoldED can only be used via the ↩ keyboard. These functions are descibed below. Other keyboard functions have associated menus but the keyboard version usually is faster (the menus tend to be unrepsonsive because the OS doesn't support setting the input queue for menu shortcuts). This manual describes the default keyboard setup. The keyboard configuration can be changed with the

Configuration/Keyboard

dialog.

Keys often 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.

1.151 Cursor keys

Cursor keys

CURSOR UP/DOWN + ALT

This sequence starts fast scrolling (up or down). The cursor position is not affected by this command, ie. 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 or the previous 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 window borders.

LEFT/RIGHT + CTRL

Indents the marked text (
Block/Mark Text
). Hold down the SHIFT key
simultaneously to indent to the next tabstop.

LEFT/RIGHT + SHIFT

Moves the cursor to the beginning of the next or the beginning of the previous word.

LEFT/RIGHT + SHIFT + ALT

Moves the cursor to the end of the next word or to the end of the previous word.

LEFT/RIGHT + CTRL + ALT

Changes the indention of the cursor line to the next tabstop.

1.152 HELP key

```
                                                  HELP key
HELP

  The HELP key folds or unfolds text: a fold is unfoldded if  the  cursor  is
  placed  over  a fold header (see
               Folding
              ), otherwise the text between the
  fold markers (if there are fold markers) is folded.

HELP + CTRL

  Folds or unfolds all folds: All folded sections are unfolded if the  cursor
  is over a fold header (see
               Folding
              ) when this function is used, otherwise
  all lines enclosed by fold markers are folded.
```

1.153 TAB key

```
                                                  TAB key
TAB (+ SHIFT)

  The TAB key (->|) moves the cursor to next tabstop.  This  editor  supports
  various TAB modes. Please see
               Tabulators
               for a detailed descriptiption of
  TAB support in GoldED. Use the TAB key with SHIFT to move the cursor to the
  previous tabstop (BACKTAB function).
```

1.154 RETURN key

```
                                                 RETURN key
RETURN (+ SHIFT)

  The RETURN key moves the cursor to the next line. The new  cursor  position
  depends  on  various settings of GoldED (see
               Configuration/User Interface
                and
               Configuration/Context
              ). Usually, the cursor is  moved  to  the  first
  column of the next line.
```

The cursor line is splitted at the cursor position if this command is used in the middle of a line (and if you do not simultaneously hold down the SHIFT key).

RETURN + CTRL

Inserts an empty line before the cursor line.

1.155 DEL key

DEL key

DELETE (+ SHIFT)

The DEL key deletes the character under the cursor and moves the rest of line one position to the left. The rest ofd the line is deleted if you simultaneously press the SHIFT key.

DELETE + CTRL

Deletes the current line. You can recall up to 50 lines deleted by this function with the Extras/Line Insert menu.

DELETE + ALT

Deletes the word under the cursor. You can recall up to 100 words deleted by this function (DEL-ALT-SHIFT).

DELETE + ALT + SHIFT

Reinsert a deleted word (see DELETE-ALT)

1.156 ESC key

ESC key

APC

APC stands for AutoCompletion: this function replaces the word under the cursor with a word from the user dictionary (Configuration/Dictionary). Example: type 'TIG', then press the ESC key. 'TIG' is replaced by 'TAG_IGNORE' if the C-dictionary is loaded. GoldED uses a pattern matching algorithm to find dictionary entries: you can use other abbreviations than 'TIG' (e.g. 'TAGI') as long as the first letter of the word to be completed matches 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 want to use
automatic completion without a dictionary.

1.157 SPACE key

SPACE key

IntelliSense

Unlike the
APC
function, the IntelliSense feature can complete words to the full form without using a dictionary. Instead, it completes words based on the context, ie. by searching a possible expanded form directly in the document. Type the first few letters of the word, then use 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 keyboard configuration of the space key. The trigger parameter determines how thoroughly the document is searched. Default is 4. Useful values range from 1 to 10. The smaller, the faster. 4 means that GoldED will stop searching for an expanded form if it has found a possible match which is at least four characters longer than the unexpanded form.

1.158 F-Keys

F-Keys

Function keys are configurable: use the
Configuration/Keyboard
dialog to
configure the keyboard. Some commodities consume all F-keys (including the F-key commodity): GoldED can not recognize function keys if such a commodity is running on your computer, ie. you will then not be able to configure these keys in GoldED.

```
          Key     Description       See
          ------------------------------------------------------------
          F1      Open  file
                 Project/Open
                          F2      Merge file
                 Project/Insert File
                          F3      Print file
```

```
                Project/Print
                         F4      Hide block
                Block/Marker Off
                         F5      Mark line
                Block/Mark Lines
                         F6      Find next
                Search/Search Next
                         F7      Next page
                Cursor Keys
                         F8      Set insert mode
                Layout/Insert Mode
                         F9      ASCII table
                Extras/Insert Special Character
                         F10     Play macro
                Macros/Sequence play
                         Key +
         SHIFT   Key Description       See
         -----------------------------------------------------------
         F1      Save as
                Project/Save As
                         F2      Save as XPK
                Project/Save Compressed File
                         F3      Quit
                Project/Close Window
                         F4      Clear text
                Project/Clear Document
                         F5      Mark line
                Block/Mark Lines
                         F6      Replace
                Search/Replace
                         F7      Page up
                Cursor Keys
                         F8      Overwrite mode
                Layout/Insert Mode
                         F9      Project setup
                Configuration/Project
                         F10     Record macro
                Macros/Sequence Record
```

1.159 Rexx port

Rexx port

Rexx macros vs. recorded sequences

GoldED supports two methods of automated control: Rexx scripts and recorded sequences. Rexx scripts are programs similar to programming languages like BASIC. They are started by GoldED (examples:
Macros/Macros
) but are
actually executed by the Rexx server RexxMast (part of the Amiga operating system, usually started in the startup-sequence). The Rexx server will run the script line-by-line and send messages to GoldED if it detects commands it doesn’t understand (commands to be handled by GoldED). Rexx is a very

powerful method to automate control. However, this approach requires at least some basic knowledge of Rexx. If you need automated control but are not interested in Rexx you can use GoldED's ability to record command sequences: Enable recording mode (Macros/Sequence Record) and perform a sequence of commands. Make the editor "learn" how to do it. Once you have recorded a sequence you can replay it as often as you like (see Macros/Sequence Play). You can save sequences Macros/Save Sequence) or assign them to events like menus or keystrokes.

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

Rexx basics

It is important that Rexx 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

1.160 Select a host

Select a host

If you use GoldED to start macros (e.g. if you create menu items of type Rexx or if you execute a document as macro with the Macros/Start Text As Macro menu), script commands which are not part of the Rexx language are sent to the editor (the "host") automatically. However, if your script is started by a different program (e.g. rx), the macro will have to choose a host (GoldED's Rexx port) for communication by using the ADDRESS command of the Rexx language: ADDRESS <port name>.

Port name

The name of GoldED's rexx port is "GOLDED.1" if the editor is run only once. The port of a further editor would be "GOLDED.2", etc. Use the About GoldED menu or use the

QUERY
command (
Macros/Execute Command
: QUERY HOST) if you
want to know the name of the Rexx port or the screen name. Most users will never run more than one GoldED task, so assuming that the port name is "GOLDED.1" should work in most cases.

1.161 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
commands 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 scripts 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 Rexx to solve this problem. In interactive mode, results of the
QUERY
command are displayed in a window (e.g. LINE=3). In Rexx ↩
mode (after a

LOCK
), results are not displayed but written to the variable RESULT ↩
without
prefixed variable name (e.g. 3).

1.162 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 send from Rexx to GoldED are parsed twice: first by the Rexx 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 Rexx 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 Rexx variable RESULT (provided that you have asked for results using the Rexx 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 sometimes are available in the special variable RC2.

1.163 Unlock GUI

Unlock GUI

A Rexx 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 Rexx (e.g. SIGNAL or OPTION FAILAT). Just have a look at the script examples in the golded:etc/rexx drawer.

Tip:

As a last resort - editor blocked by erroneous macro - you can use the function "Unlock editor" from the Drag & Drop program (in the the GoldED drawer) to unlock the editor.

1.164 Internal commands

Internal commands

GoldED offers about 250 commands and options (see
Command list
), supported
by all interfaces of GoldED: you can use them in Rexx macros, bind them to menus (
Configuration/Menus

), attach them to keys (
Configuration/Keyboard
)
or execute them directly (
Macros/Execute Command
). AmigaDOS rules apply as
far as the syntax is concerned because GoldED uses the ReadArgs() function of the Amiga operating system to parse commands: strings containing spaces must be quoted, options and keywords can be uppercase or lowercase. The line below is an example for a syntax descriptions as 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 (/N), other introduce strings (/K), etc. Most options can be used simultaneously in a single command line. Options followed by "/A" may not be omitted.

The options FORCE and ALL are switches (/S = "switch"): they make the print command behave in a special way (described in this manual) if these option are used in the command line. No further arguments are expected. Example:

```
PRINT ALL
```

The keyword ITALICS introduces a keyword/value pair (/K = "keyword"): an argument is expected after the keyword. The argument type depends on the context 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 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).

1.165 Command List

Command List (use: see
Internal Commands
):

API

DPAGE

HELP

MOUSE

QUIT

TAB

BACK

ELSE

HUNTER

NAME

REDO

TABS

BEEP

ENDIF

IF

NEW

REFRESH

TASK

BIND

ENDWORD

IMAGES

NEXT

REGEDIT

TEXT

BITS

EXALL

INDENT

NOP

REMAP

TMPLATE

BLOCK

EXPAND

INFO

NOTIFY

REPLACE

TOOLBAR

BRACKET

EXTRACT

INSERT

OPEN

REQLIST

TYPE

BREAKPT

FDOWN

KEY

PATH

REQUEST

UJUMP

CLIP

FILE

LAYOUT

PC

RIGHT

UNDO

CMD

FIND

LEFT

```
PHRASE

RUN

UNLOCK

CODE

FIRST

LINES

PING

RX

UP

COLON

FIX

LOCK

PONG

SAVE

UPAGE

CONTEXT

FOLD

MACRO

POP

SCREEN

USE

CR

FORMAT

MAN

PREFS

SESSION

VIEW

DEBUG
```

```
FREEZE

MARK

PREV

SET

VLEFT

DEL

FUNC

MAXDOWN

PREVEND

SHIFT

VRIGHT

DELETE

FUP

MAXUP

PRINT

SMARTCR

WINDOW

DIR

GOTO

MENUS

PROJECT

STOP

WORD

DJUMP

GREP

MISC

PUSH

SUFFIX

XREF
```

```
                 DOWN

                 GUI

                 MORE

                 QUERY

                 SYNTAX
```

1.166 API

```
               API

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  API        ASK/S         open API configuration requester
```

1.167 BACK

```
               BACK

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  BACK       (no options)  backspace operation
             SMART/S       backspace over marked word deletes word
```

1.168 BEEP

```
               BEEP

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  BEEP       (no options)  display beep
```

1.169 BIND

```
              BIND

  Description of
                 Internal Commands
                  command    option        description
  ---------------------------------------------------------------------------
  BIND       ASK/S         open keyboard configuration requester
```

1.170 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)
```

1.171 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

             Comment: A block can be deleted with
                 DELETE
```

1.172 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. "()")
```

1.173 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.
```

1.174 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
```

1.175 CMD

```
              CMD

  Description of
               Internal Commands
                command    option        description
 ---------------------------------------------------------------------------
 CMD        ASK/S         open command requester
            COMMAND/K     command to be executed (STRING)
```

1.176 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.
```

1.177 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.
```

1.178 CONTEXT

```
             CONTEXT

 Description of
              Internal Commands
               command    option        description
 ---------------------------------------------------------------------------
 CONTEXT    MENU/S        Show context menu
            RELMOUSE/S    Context menu position depends on mouse pointer
```

1.179 CR

```
             CR

 Description of
              Internal Commands
               command    option        description
 ---------------------------------------------------------------------------
 CR          (no options)  <RETURN> command; splits line at cursor position

            Comment: see
                Configuration/Context
```

1.180 DEBUG

```
             DEBUG

 Description of
              Internal Commands
               command    option        description
 ---------------------------------------------------------------------------
 DEBUG      START/S       start debugger support
            STOP/S        stop  debugger support
            PORT/K        Rexx port of debugger (STRING)
            RESET/S       Reset breakpoints (query breakpoints for all buffers)

            Comment: This command is reserved for usage by a debugger.
```

1.181 DEL

```
               DEL

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  DEL        (no options)  deletes character under cursor
```

1.182 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).
```

1.183 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/Set Path
```

1.184 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
              .
```

1.185 DOWN

```
              DOWN

Description of
               Internal Commands
                command    option        description
-----------------------------------------------------------------------------
DOWN        (no options)  move cursor one line down
```

1.186 DPAGE

```
              DPAGE

Description of
               Internal Commands
                command    option        description
-----------------------------------------------------------------------------
DPAGE       (no options)  show next page (compare:
               DJUMP
              )
```

1.187 ELSE

```
              ELSE

Description of
               Internal Commands
```

```
                command    option        description
---------------------------------------------------------------------------
ELSE       (no options)  introduces alternative
               IF
               section
```

1.188 ENDIF

```
              ENDIF

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
ENDIF      (no options)  terminates
               IF
```

1.189 ENDWORD

```
              ENDWORD

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
ENDWORD    (no options)  move cursor to end of word
```

1.190 EXALL

```
              EXALL

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
EXALL      (no options)  examine text

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

1.191 EXPAND

```
               EXPAND

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  EXPAND     VAR/K         Rexx variable to hold the result (STRING)
             NAME/K        path (STRING)

             Comments: Expands given path (logical assigns are resolved).
```

1.192 EXTRACT

```
               EXTRACT

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  EXTRACT    (no options)  extract file name under cursor
             VAR/K         where to put the result: Rexx 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.
```

1.193 FDOWN

```
               FDOWN

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  FDOWN      (no options)  scroll down (fast)
```

1.194 FILE

```
               FILE

  Description of
                Internal Commands
                command    option        description
----------------------------------------------------------------------------
FILE       NAME/K        file/directory name (STRING)
           DELETE/S      delete file
           RENAME/S      rename file
           INFO/D        examine file (protection bits and size)
           FORCE/S       don't ask for confirmation
           SEARCH/K      search this path for specified file (STRING)
           VAR/K         Rexx variable to hold the result (STRING)
           NEWDIR/S      create directory
           TOUCH/S       reset file date

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

1.195 FIND

```
              FIND

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
FIND       STRING/K      search pattern (STRING)
           THIS/S        use word under cursor as search pattern
           MARKED/S      use marked text as search pattern
           INDEX/S       list found strings in a requester
           NEXT/S        find next occurrence
           PREV/S        find previous occurrence
           COUNT/S       count pattern (doesn't affect cursor position)
           BLOCK/S       count in block only
           WILD/K        set wildcard mode (BOOL)
           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)
```

1.196 FIRST

```
              FIRST

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
FIRST      (no options)  move to beginning of line (see
               GOTO
              )
```

1.197 FIX

```
              FIX

  Description of
               Internal Commands
                command    option        description
  ---------------------------------------------------------------------------
  FIX        VAR/K/A       Rexx variable name (STRING)

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

1.198 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)
```

1.199 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
```

1.200 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
```

1.201 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
```

1.202 FUP

```
                FUP

  Description of
                 Internal Commands
                  command    option        description
  ---------------------------------------------------------------------------
  FUP        (no options)  scroll backwards (fast)
```

1.203 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
                           BYTE/N        byte offset to go to (ULONG)
           UNFOLD/K      unfold if necessary ? (BOOL)
           OFFSET/S      go to byte offset
           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
           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  one line in
           UNFOLD=FALSE mode.
```

1.204 GREP

```
              GREP

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
GREP       STRING/K      string to search project files for (STRING)
           ASK/S         show grep requester
           UPDATE/S      rescan files (to be used with ASK/S)
           NEW/S         request a pattern (to be used with ASK/S)
           CASE/K        case sensitive search ? (BOOL)
           WORDS/K       look for whole words only ? (BOOL)
```

1.205 GUI

```
              GUI

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
GUI        ASK/S         open GUI configuration window
```

1.206 HELP

```
                HELP

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

             Comment: The menu's guide is used (
                 Configuration/Menus
                ) if  no
             guide is specified.
```

1.207 HUNTER

```
                HUNTER

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

1.208 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.
```

1.209 IMAGES

```
               IMAGES

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
IMAGES     VALIDATE/S    validates the image cache (based on file dates)
           RESET/S       clears the image cache
```

1.210 INDENT

```
               INDENT

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
INDENT     ASK/S         open indention configuration requester
```

1.211 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
```

1.212 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 a file name (chosen from file requester)
             DIR/S         insert a drawer name (chosen from file requester)
```

1.213 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
```

1.214 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.
```

1.215 LEFT

```
              LEFT

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
LEFT       (no options)  move cursor one position to the left
```

1.216 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
```

1.217 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)
           SEQUENCE/S    execute macro in sequence mode

           Comment: To be used in  Rexx  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 command moves GoldED's screen to the front.

           The SEQUENCE option suppresses dialog windows (assuming that the
           macro  has  been  generated by exporting a recorded sequence and
           that it does contain all necessary user input).
```

1.218 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)
           LOAD/S        load a sequence
           SAVE/S        save recorded sequence
           REXX/S        save as Rexx macro (to be used with SAVE/S)
           REPEAT/S      repeat the next event (keystroke, menu, etc.)

           Comment: Sequence playback  is  disabled  during  execution  of
           Rexx macros.
```

1.219 MAN

```
             MAN

Description of
              Internal Commands
               command    option        description
---------------------------------------------------------------------------
MAN        ASK/S         shows the references requester
           BUILD/S       rebuilds als databases
```

1.220 MARK

```
             MARK

Description of
              Internal Commands
               command    option        description
---------------------------------------------------------------------------
MARK       HIDE/S        hide mark
           PARAGRAPHE/S  mark paragraph
           WORD/S        mark word under cursor
```

```
             ALL/S         mark all lines
             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        marker mode: lines
             COLUMN/S      marker mode: characters
             VERTICAL/S    marker mode: columns
             STRICT/S      only SPC (ASCII 32) is regarded as word delimiter
```

1.221 MAXDOWN

```
                MAXDOWN

  Description of
                 Internal Commands
                  command    option        description
  ---------------------------------------------------------------------------
  MAXDOWN    (no options)  move cursor to next quarter of document
```

1.222 MAXUP

```
                MAXUP

  Description of
                 Internal Commands
                  command    option        description
  ---------------------------------------------------------------------------
  MAXUP      (no options)  move cursor to previous quarter of document
```

1.223 MENUS

```
                MENUS

  Description of
                 Internal Commands
                  command    option        description
  ---------------------------------------------------------------------------
  MENUS      ASK/S         open menu configuration requester
```

1.224 MISC

```
                MISC

  Description of
                 Internal Commands
                  command    option        description
  ---------------------------------------------------------------------------
  MISC       ASK/S         open misc configuration requester
```

1.225 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
```

1.226 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        marker mode: lines   (default: characters)
             VERTICAL/S    marker mode: columns (default: characters)
             DRAG/S        activate drag & drop support

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

1.227 NAME

```
                NAME
```

```
Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
NAME       ASK/S         display text name requester
           NEW/F         set new text name (STRING)
```

1.228 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'
           LINES/N       buffer preallocation request (ULONG)

           Comment: LINES/N can be used to preallocate memory  for  storing
           the  specified  number  of lines. Usage of this option should be
           avoided since built-in dynamic allocation is more effective.
```

1.229 NEXT

```
              NEXT

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
NEXT       (no options)  move cursor to next word
```

1.230 NOP

```
              NOP

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
NOP        (no options)  this command does nothing (no operation)
```

1.231 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.
```

1.232 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)
             SUGGEST/K     fix spelling errors in file names ? (BOOL)
             TOFRONT/K     bring window to the front ? (BOOL)

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

1.233 PATH

```
             PATH

Description of
              Internal Commands
               command    option        description
---------------------------------------------------------------------------
PATH       ASK/S         open hunter configuration requester
```

1.234 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.
```

1.235 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)
```

1.236 PING

```
                PING

  Description of
                 Internal Commands
                  command    option        description
 ---------------------------------------------------------------------------
  PING       SLOT/N        save cursor position to slot (UWORD: 0...9)

             Comment: see
                 PONG
```

1.237 PONG

```
                PONG

  Description of
                 Internal Commands
                  command    option        description
 ---------------------------------------------------------------------------
  PONG       SLOT/A/N      recall cursor positions (UWORD 0-9)

             Comment: GoldED offers eight bookmark storages  (0-7)  for  each
             window  (see
                 PING
                ).  Slot  0  is  reserved  for usage in Rexx
             scripts. Bookmarks move with the text.
```

1.238 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.
```

1.239 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, ...)
```

1.240 PREV

```
              PREV

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
PREV       (no options)  move cursor to previous word
```

1.241 PREVEND

```
              PREVEND

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
PREVEND    (no options)  move cursor to end of previous word
```

1.242 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
```

1.243 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 *)
```

1.244 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
```

1.245 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  a  Rexx
             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      automatically fold files after loading ?
ALEFT                BOOL      layout: use current indention ?
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 ?
AUTOINDENT           BOOL      keep indention after CR ?
BACKUP               NUMBER    number of backups generations
BAKDIR               STRING    backup directory
BAKPREFIX   READONLY STRING    backup file prefix
BITS                 STRING    protection bits
BLOCK       READONLY NUMBER    block type (0=none 1=lines 2=text 3=column)
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 ?
CLASS       READONLY STRING    Syntax level of char under cursor; see LEVEL
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
CONTAINER            BOOL      show plug-in containers ?
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
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
```

```
HIDE                 BOOL      error requesters disabled ?
HMI                  NUMBER    printer line feed (0 - 2)
HOST        READONLY STRING    Rexx 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 char under cursor; see CLASS
LINE        READONLY NUMBER    line number (relative); ABSLINE
LINES       READONLY NUMBER    lines (relative); see ABSLINES
LOGIN                STRING    login name
LPI                  NUMBER    lines/inch: 6 lpi or 8 lpi (UWORD: 0, 1)
LQ                   BOOL      letter quality printing ?
MAKE                 STRING    make macro (project management)
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
POSITION    READONLY STRING    mouse position description
PREVIEW              BOOL      preview active ?
PRJLIST     READONLY NUMBER    project data (see developer/source/project)
PROG        READONLY STRING    program name
RC          READONLY NUMBER    error code of last command
READONLY             BOOL      document write protected ?
RECORD               BOOL      sequence recording activated ?
REFORMAT             BOOL      reformat while typing ?
REM                  STRING    file comment
RESET                BOOL      reset printer before output ?
RESIDENT             BOOL      load editor resident ?
RESULT      READONLY STRING    result string of last command
RIGHT                NUMBER    layout: right border
ROWS        READONLY NUMBER    windo>w height; see WINH
RPLC                 STRING    replace text
RXDEBUG              BOOL      output Rexx 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 indention ?
SPC                  STRING    white space definition string
STDLINE     READONLY BOOL      current line not write-protected ?
STYLE                NUMBER    layout style for online formatting (0...3)
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)
TRACE                 BOOL      show Rexx debug information ?
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      word wrap 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.
```

1.246 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
```

1.247 REDO

```
              REDO

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
```

```
REDO       LAST/S        undo last
               undo
```

1.248 REFRESH

```
              REFRESH

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
REFRESH    PAGE/S        redraw current text
           LINE/S        redraw current line
```

1.249 REGEDIT

```
              REGEDIT

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
REGEDIT                  the REGEDIT command should not be used
```

1.250 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
```

1.251 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.
```

1.252 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         Rexx variable to hold the result (STRING)

           Comment: Presents a listview. Selected string is returned.
```

1.253 REQUEST

```
              REQUEST

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
REQUEST    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
           FILES/S       ask for files (multi selection)
           DIR/S         ask for a directory
           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         Rexx variable to hold 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)
            NEXT/S        activate next requester

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

1.254 RIGHT

```
               RIGHT

  Description of
                Internal Commands
                 command    option        description
 ---------------------------------------------------------------------------
 RIGHT      (no options)  move cursor one position to the right
```

1.255 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)
```

1.256 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 the Rexx server (STRING)
           MACRO/K       macro to execute if PORT is not valid (STRING)
           TEXT/S        start document as macro

           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 REXX: Commands in '...' are interpreted as commands,
           commands in "..." are interpreted as macro names.
```

1.257 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.
```

1.258 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
           UNDEF/S       use standard screen size

           Comment: Screen and overscan dimensions are  not  saved  to  the
           display configuration file is UNDEF is specified.
```

1.259 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
```

1.260 SET

```
              SET

Description of
               Internal Commands
                command    option        description
---------------------------------------------------------------------------
SET        NAME/K        variable to be set (STRING):
               Variables
                           VALUE/K       new value (STRING)
```

1.261 SHIFT

```
              SHIFT

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

1.262 SMARTCR

```
              SMARTCR

  Description of
               Internal Commands
                command    option        description
  ---------------------------------------------------------------------------
  SMARTCR    (no options)  return (line is not splitted at cursor position)
```

1.263 STOP

```
              STOP

  Description of
               Internal Commands
                command    option        description
  ---------------------------------------------------------------------------
  STOP       (no options)  stop command execution
```

1.264 SUFFIX

```
              SUFFIX

  Description of
               Internal Commands
                command    option        description
  ---------------------------------------------------------------------------
  SUFFIX     STRING/K      file name (STRING)
             SUFFIX/K      desired suffix (STRING, e.g. ".c")
```

1.265 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
```

1.266 TAB

```
TAB

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

1.267 TABS

```
                TABS

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  TABS       ASK/S         open tab configuration requester
```

1.268 TASK

```
                TASK

  Description of
                Internal Commands
                 command    option        description
  ---------------------------------------------------------------------------
  TASK       WAIT/N        wait interval (1/50 sec) (UWORD)
             BENCH/S       run graphics benchmark
```

1.269 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 "**".
```

1.270 TMPLATE

```
               TMPLATE

 Description of
                Internal Commands
                 command    option        description
 ----------------------------------------------------------------------------
 TMPLATE    ASK/S         open templates configuration requester
            CHECK/S       check word under cursor
```

1.271 TOOLBAR

```
               TOOLBAR

 Description of
                Internal Commands
                 command    option        description
 ----------------------------------------------------------------------------
 TOOLBAR    ASK/S         open toolbar configuration requester
            SHOW/K        show/hide toolbar (BOOL)
            NAME/K        toolbar name (STRING)
```

1.272 TYPE

```
               TYPE

 Description of
                Internal Commands
                 command    option        description
 ----------------------------------------------------------------------------
 TYPE       RESET/S       autodetect filetype of current text
```

1.273 UJUMP

```
               UJUMP

 Description of
                Internal Commands
                 command    option        description
```

```
----------------------------------------------------------------------------
UJUMP      (no options)  jump to beginning of window/previous page

           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
              .
```

1.274 UNDO

```
              UNDO

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

1.275 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

           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.
```

1.276 UP

```
              UP

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
```

```
UP         (no options)  move cursor one line up
```

1.277 UPAGE

```
              UPAGE

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
UPAGE      (no options)  show previous page (compare
               UJUMP
              )
```

1.278 USE

```
              USE

Description of
               Internal Commands
                command    option        description
----------------------------------------------------------------------------
USE        (no options)  accept current line

           Comment: To be  used  within  Rexx  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.
```

1.279 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.

1.280 VLEFT

VLEFT

Description of
Internal Commands

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

1.281 VRIGHT

VRIGHT

Description of
Internal Commands

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

1.282 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).
```

1.283 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
```

1.284 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 ?
```

1.285 Input events

Input events

Input event insertion

GoldED’s KEY command can be used to insert simulated "input events" (keystrokes) into intuition’s global input stream. Inserting events makes the current application (usually the active GoldED window) behave as if the input events were generated by the user. 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 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 simulate the return key). Syntax for description strings: <CLASS QUALIFIER(S) KEY>

A) CLASS may be one of the following (default is <rawkey>)

rawkey	this is a keyboard event
rawmouse	this is a mouse button event

B) QUALIFIER(s) may be one or more of these:

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

C) KEY may either be a plain character or one of these:

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 rcomand o>"
```

1.286 RECOVER

RECOVER

Tools

The recover program in the GoldED drawer can be used to recover text buffers after your system has crashed. Recover will scan all available memory locations (the free memory pool and 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 difficult 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 (the recover program defaults to ignoring a text if it has not been modified).

TEST

Just scan memory for text buffers to recover but do not 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 finds a longer line while scanning memory it will refuse to restore the line because it probably is corrupt. Default is 255 characters.

MAXLINES

Maximum number of lines (per text) to be restored. Recover will stop recovery if a text exceeds this limit (text is probably corrupt). Default is 10000 lines.

1.287 ORDER

ORDER

You can purchase this software at your local Amiga dealer or directly from the developer. Please visit the web site (WWW Support) for pricing information.

1.288 UPDATES

UPDATES

Minor updates usually are free and can be downloaded from the web site (see WWW Support). Major updates and huge service packs are released on CD-R/CD-ROM and can be purchased directly from the developer. Please visit the web site for pricing information. You will have to return the GoldED registration card before you can order updates.

1.289 CREDITS

CREDITS

* DICE * Reqtools * XPK * ARexxBox * GadToolsBox *

The work of many people has gone into creating GoldED. I want to give credit to those who've helped me with the development (and hopefully I'm not forgetting anybody): Thanks to Nico François for his reqtools library and to the developers of the XPK compression standard. The Rexx routines of GoldED have been inspired by source code created by Michael Balzer. Thanks to Stefan Zeiger for Boopsi example source code. And thanks to Joerg Gutzke, Dario Fava &amp; Thomas Lechner, sysops in Aachen who've helped me distributing the early versions of GoldED. GUIMake has been developed by Rico Krasowsk. Finally, I would like to thank these people for their suggestions, translations, ideas &amp; support: Giovanni Addabbo, Henric Andersson, Markus Aretz, Olaf Barthel, Jochen Becher, Thomas Bliesener, Cristian Castellari, Ernesto Poveda Cortes, Martin Fay, David Göhler, Georges Goncalves, Christian Gottschling, Llorenç Grau, Andreas Harrenberg, Mick Hohmann, Henning Hucke, René Laederach, Lieven Lema, Marcin Orlowski, Lars Renström, Rodolphe Sanderson, Stefan Schor, Oleg Sergeev, Maarten Ter Mors and Kim Roar Utsi. Further credits goes to the following GoldED users who created many useful Rexx scripts: Eric Burghard, Oliver Clouth, Leu Simon Gris, Francois Helsen, Tattoo Mabonzo, Krzysztof P. Jasiutowicz, Fin Schuppenhauer and Markus Zahn. And last but not least I wish to thank those of you who have supported the development simply by registering GoldED - this program would not have been possible without you !

1.290 ADDRESS

ADDRESS

Bugreports, comments and suggestions are welcome. However, personal technical support is available for registered users only. You must have returned the GoldED registration card before you request support. Please state the serial code of your editor when contacting the developer. Users of a free GoldED trial version - while not having access to personal support - can find some GoldED-related information at the WWW site (information about new versions, etc.).

WWW SUPPORT

Visit the internet support site to find the lastest news, updates and tools:

http://members.tripod.com/golded

DEVELOPER ADDRESS

```
  Dietmar Eilert
  Mies-v-d-Rohe-Str.31, 52074 Aachen, Germany
  Phone: +49-(0)179-5987061 German/English
  E-Mail: Dietmar.Eilert@post.rwth-aachen.de
  E-Mail: dietmar_eilert@yahoo.de (alternative address)
  WWW:    http://members.tripod.com/golded
  Mirror: http://members.xoom.com/golded
```

1.291 GoldED

ABOUT GOLDED

FIND

PROJECT/ICONIFY

ADDRESS

FIRST

PROJECT/INSERT FILE

APC

FIX

PROJECT/OPEN

API

FOLD

PROJECT/OPEN IN NEW WINDO

AREXXBOX

FOLDING

PROJECT/OPEN ORIGINAL

ARGUMENTS

FORMAT

PROJECT/PRINT

AUTOBACKUP

FREEZE

PROJECT/PROPERTIES

AUTOCASE

FUNC

PROJECT/SAVE

BACK

FUP

PROJECT/SAVE AND CLOSE

BEEP

GADTOOLSBOX

PROJECT/SAVE AS

BIND

GETTING STARTED

PROJECT/SAVE COMPRESSED F

BITS

GLOBAL SEARCH

PROJECT/SET PATH

BLOCK

GOTO

PROJECT/USE CURRENT PATH

BLOCK MENU

GREP

PUSH

BLOCK/BLOCK COPY

GUI

QUERY

BLOCK/BLOCK DELETE

HELP

QUICKFUNC

BLOCK/BLOCK MOVE

HELP KEY

QUICKREFERENCE

BLOCK/CLIPBOARD COPY

HISTORY

QUICKSTARTER

BLOCK/CLIPBOARD CUT

HOTKEY

QUIT

BLOCK/CLIPBOARD PASTE

HOW TO FOLD LINES

RECOVER

BLOCK/CONVERT TO LOWERCAS

HUNTER

REDO

REXX PORT

BRACKET

KEY

RIGHT

BREAKPT

KEYBOARD

RUN

CHARACTER SET REMAPPING

LANGUAGE

RX

CLIP

LAYOUT

SAVE

CLIPBOARD

LAYOUT MENU

SCREEN

CMD

LAYOUT/CORRECT CASE

SEARCH MENU

CODE

LAYOUT/DETECT TEMPLATES

SEARCH/CHECK NESTING

COLON

LAYOUT/FORMAT BLOCK

SEARCH/COUNT PATTERN

COMMAND LIST

LAYOUT/FORMAT BLOCK BLOCK

SEARCH/FIND FILE

```
CONFIGURATION

LAYOUT/FORMAT BLOCK CENTE

SEARCH/FIND FILE...

CONFIGURATION/CONTEXT

LAYOUT/FORMAT BLOCK LEFT

SEARCH/FIND IN FILES

CONFIGURATION/DICTIONARY

LAYOUT/FORMAT BLOCK RIGHT

SEARCH/REFERENCE

CONFIGURATION/DISPLAY

LAYOUT/FORMAT PARAGRAPH

SEARCH/REFERENCE...

CONFIGURATION/DISPLAY/COL

LAYOUT/FORMAT PARAGRAPH B

SEARCH/REPLACE

CONFIGURATION/DISPLAY/DET

LAYOUT/FORMAT PARAGRAPH C

SEARCH/REPLACE NEXT

CONFIGURATION/DISPLAY/FON

LAYOUT/FORMAT PARAGRAPH L

SEARCH/SEARCH

CONFIGURATION/DISPLAY/SCR

LAYOUT/FORMAT PARAGRAPH R

SEARCH/SEARCH BACKWARDS

CONFIGURATION/DISPLAY/WIN

LAYOUT/INDENT BLOCK

SEARCH/SEARCH NEXT

CONFIGURATION/FILE SEARCH

LAYOUT/INDENT LINE
```

TAB KEY

CONTEXT

MACROS/SAVE AS REXX MACRO

TABS

CONTEXT MENUS

MACROS/SAVE SEQUENCE

TABULATORS

CR

MACROS/SAVE SESSION

TASK

CREDITS

MACROS/SEQUENCE APPLY TO

TEMPLATES

CURSOR KEYS

MACROS/SEQUENCE PLAY

TEXT

DEBUG

MACROS/SEQUENCE PLAY LOOP

TMPLATE

DEBUGGER

MACROS/SEQUENCE RECORD

TOOLBAR

DEL

MACROS/START TEXT AS MACR

TYPE

DEL KEY

MAGIC CODES

UJUMP

EXTRAS/COMPLETE TEXT

MOUSE

VIEW/GO TO OFFSET

EXTRAS/COMPLETE TEXT/COMP

MOUSE BUTTONS

VIEW/HIDDEN DOCUMENTS

EXTRAS/COMPLETE TEXT/COMP

MOUSE CONFIGURATION

VIEW/OPEN NEW WINDOW

EXTRAS/CUSTOMIZE

MULTISELECTION

VIEW/RECALL POSITION

EXTRAS/CUSTOMIZE/FILETYPE

NAME

VIEW/SHOW BREAKPOINTS

EXTRAS/CUSTOMIZE/GLOBAL O

NEW

VIEW/SHOW COLORS

EXTRAS/CUSTOMIZE/GLOBAL O

NEXT

VIEW/SHOW PLUG-INS

EXTRAS/CUSTOMIZE/GLOBAL O

NOP

VIEW/SHOW PREVIEW

EXTRAS/CUSTOMIZE/GLOBAL O

NOTIFY

VIEW/SHOW TOOLBARS

EXTRAS/CUSTOMIZE/GLOBAL O

ONLY WHOLE WORDS

EXTRAS/OPEN SHELL

POP

VRIGHT

EXTRAS/REDO

PREEVIEW

WHITE SPACE

EXTRAS/RENAME FILE

PREFS

WILDCARDS

EXTRAS/SAVE SETTINGS

PREV

WINDOW

EXTRAS/SWAP ADJACENT LINE

PREVEND

WORD

EXTRAS/UNDO

PRINT

WORD WRAP

F-KEYS

PROJECT

WWW SUPPORT

FDOWN

PROJECT MENU

XPK

FILE

PROJECT/APPEND FILE

XPK SUPPORT

FILE LIST

PROJECT/CLEAR DOCUMENT

XREF

FILETYPES

PROJECT/CLOSE WINDOW

FIND

PROJECT/EXIT EDITOR