Documentation for GED-Hexedit

COLLABORATORS			
	TITLE : Documentation for GED-Hexedit		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 9, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Documentation for GED-Hexedit

1.1 Documentation for GED-Hexedit

```
                              GED-Hexedit Version 1.00 by Wolfgang Morgeneier
```

1.2 Introduction

Introduction:

For the Amiga there are not many hex-editors. Some day I found a program <BinHex> in the Aminet. After seeing this, I had the idea, to write a similar program and an environment for hexadecimal edit for the editor GOLDED by Dietmar Eilert. This here is the preliminary finished result of it.

This program can have bugs!
I will not be responsible for any bugs of the program or loss of data!

To avoid this, it is best to work only with copies of your files!

If somebody is interested in a further development of the program by me, he should send me an EMail with error reports or ideas for further development.

1.3 Installation

Installation of the program:

The program will be installed by the Amiga-installer.
For this, the icon <Install_English> has to be clicked twice.
After this the installation goes automatic.

The following is required for the program:
- Golded 4.6.0+ by Dietmar Eilert
- Regedit (for the installation)
- the Amiga-installer in the search-path (for the installation)
- delete, copy and rename in directory c:
- Arexx
- the directory t: must exist
  as often files will be intermediate saved at t:, t: should be in RAM
  (if you do not have to less memory)
- a fast Amiga computer (otherwise operation is very slow)
- RAM memory (approx. 10 times the size of the file to be edited will be needed)

1.4 Usage

Usage and function of the program:

The program installs a new icon at the end of the toolbar with inscription <HEX>.
By clicking this icon, a file-requester shows up, and you can select a file
for hexadecimal edit.
The file to be edited will then be loaded into memory, reformated and shown.

The display is separated into 3 parts:
- at the left there is the address in hex format
- in the middle are the contents of the file in hex format, 16 bytes per line,
  each long word is divided by a space
- at the right there are the contents of the file in ASCII format. If a char
  cannot be displayed a <·> is shown.

Example:

```
000014B0  1F223344 44552334 55667772 22332222  ··3DDU#4Ufwr·3··
000014C0  0A112345 67890123 45677777 88890234  ··#Eg··#Egww···4
```

The keymap has been reconfigured to ensure, that only proper chars can be entered
at each place.

The address area and all spaces cannot be changed.
In the hex area, all digit keys, the letter keys a ... f and the cursor keys are allowed. The letters can be entered in capital or small spelling. They are shown always in capital spelling.
In the ASCII area additional all letters and most of the additional chars are allowed.

A special key is the tabulator key.
With this key you can jump between the HEX area and the ASCII area.

The mouse is used to place the cursor to the location you want.

Describtion of the menu

Describtion of the toolbar

1.5 Toolbar

Describtion of the toolbar

The toolbar consists of 6 buttons.
They have the following meaning (from left to right):

- New: open new window in standard mode

- Open: open an existing file in HEX mode

- Save: save the actual file

- Undo: undo the last action

- Redo: undo the last undo action

- Help: call help for program GED_Hexedit

back

1.6 Menu

Describtion of the menu

File

Window

Edit

Search

Options

Help

back

1.7 File

Menu File

- new text:
  opens a new text without name in standard mode

- open new normal:
  opens an existing file in normal mode. The file can be selected at a file requester.

-open new hex:
  opens an existing file in hex mode. The file can be selected at a file requester.

- save:
  saves the file with its existing file name.

- save as:
  shows up a file requester, where you can change the file name and saves the file.

- file properties:
  with this you can change the file name, the file comment, the protection bits and the file type. The file type of files, which are edited by GED_Hexedit, is always <HEX>.
  Note: if the display of the file is corrupted by a unforeseen event or a bug, you can repair it by deleting the file type, normal changing of the file and, after this, changing the file type again to <HEX>.

- close:
  the actual file will be closed. If it was changed a safety requester will appear before.

- quit:
  all windows will be closed and the program will quit.
  When the actual file has changed, a safety requester will appear before.
  If additional windows are open, a safety requester appears, regardless whether these files were changed.

- open files:

```
  here all open files show up and can be selected for editing.


               Back
```

1.8 Window

```
                 Menu Window


- window:
  with this menu item and its subitems you can change the size and view of the
  windows (see manual for GoldED).

- window next:
  the next open window becomes the actual text window.

- window previous:
  the previous open window becomes the actual text window.

- hidden texts:
  with this menu item and its subitems you can change the size and view of the
  windows (see manual for GoldED).


               Back
```

1.9 Edit

```
                 Menu Edit

- undo:
  undos the last action.

- redo:
  undos the last undo-action.

- save position:
  up to 5 positions can be stored.  You can jump to this positions by activating
  the next menu item.

- goto position:
  jumps to up to 5 saved text positions

- goto byte:
  with this you can jump to a specified byte in the hex area.  You have to enter
  the address in a requester.
  This can be done in 2 different ways:
```

```
  - by entering the $ sign first in the line, you can enter the address in
    hexadecimal format
  - otherwise you enter the address in decimal format

- insert byte:
  in front of the actual byte a new byte will be inserted.  Therefore the cursor
  must be in the hex area.
  ATTENTION: doing this with programs they normally cannot be started anymore.
  Therefore a safety requester appears before the action will be done.

- delete byte:
  the actual byte will be deleted.  Therefore the cursor must be in the hex area.
  ATTENTION: doing this with programs they normally cannot be started anymore.
  Therefore a safety requester appears before the action will be done.



                Back
```

1.10 Search

```
                  Menu Search


- search file:
  you can search for a file (see manual for GoldED)

- search hex bytes:
  a sequence of hex bytes will be searched for.  You have to enter the search
  string in a requester.  All digits and the letters a ... f in small or capital
  letters are allowed.  If search wildcards is on also the question mark can be
  entered.  As a maximum 16 bytes (= 32 chars) can be entered each byte-wise.
  After confirmation with <OK> the hex area will be searched in downward direction
  for the byte sequence.  If a byte sequence is found, the cursor will be put to
  the start point of it.  The entered search string will be saved and used as
  default value for the next search action.

- search again hex bytes:
  a sequence of hex bytes will be searched also.  But this cannot be entered here,
  because the last entered string will be used as search string.  This will only
  function the correct way, when the last search was a hex search.

- search ASCII:
  a sequence of ASCII chars will be searched for.  You have to enter the search
  string in a requester.  All digits, all small or capital letters, space and most
  of the special chars are allowed.  If search wildcards is on the question mark
  has the special function of a wildcard.  As a maximum 16 chars can be entered.
  After confirmation with <OK> the ASCII area will be searched in downward ↩
     direction
  for the char sequence.  If a char sequence is found, the cursor will be put to
  the start point of it.  The entered search string will be saved and used as
  default value for the next search action.

- search again ASCII:
```

a sequence of chars will be searched also. But this cannot be entered here, because the last entered string will be used as search string. This will only function the correct way, when the last search was an ASCII search.

- replace hex bytes:
  This is an extension for the function search hex bytes. When a sequence of hex bytes was found, a replacing sequence of hex bytes can be entered. With this the original byte sequence will be replaced. The replacing sequence of hex bytes must be the same length as the searched sequence of hex bytes. The question mark may not be entered as a wildcard. The sequences entered for searching and replacing will be stored and used as default value for the next action.

- replace ASCII:
  This is an extension for the function search ASCII. When a sequence of chars was found, a replacing sequence of chars can be entered. With this the original char sequence will be replaced. The replacing sequence of chars must be the same length as the searched sequence of chars. The question mark may not be entered as a wildcard. The sequences entered for searching and replacing will be stored and used as default value for the next action.

- different capital/small case:
  if this menu item is marked searching for capital letters is not the same as searching for small letters.

- with wildcard:
  if this menu item is marked, you can enter wildcards for searching. This means you can enter the char <?> instead of any char. When searching in the hex area the char <?> has to be entered for a full byte, i. e. you have to enter allways 2 question marks. Furthermore the first char and/or byte may not be a wildcard.

Back

1.11 Options

Menu Options

- Toolbar:
  switches the toolbar on or off

- local options:
  change the local options (see manual for GoldED)

- global options:
  change the global options (see manual for GoldED)

- save options:
  save the actual options (see manual for GoldED)

Back

1.12 Help

Menu Help

- Statistics:
  shows a requester with informations about the text statistics (see manual for GoldED)

- GED_Hexedit:
  shows this help file

- about GoldED 4:
  shows a requester with informations about GoldED

Back

1.13 Function describtion

Describtion of the program functions

This part deals with the internas of the progam. Reading this part is not necessary for operating the program.

The program is based on the editor GoldED by Dietmar Eilert. This editor will be changed by a new keyboard, menu, toolbar and syntax registry to work with hexadecimal datas. Furthermore most actions will be done by ARexx programs.
These will be copied during installation process to the directory GOLDED:tools/hex.

During loading a file the ARexx-Program <OpenAsHex.ged> will be called. This program first makes a safety copy of the original file (original + extension .bak.hex).
After this the program <hexbin> is called which prepares a temporary hex-coded file with name <t:gedhex> based on the original file. This file will be loaded by the editor. Now the top display of the editor will be changed to show the correct file name. After this the temporary file will be deleted.

During saving with the ARexx-program <SaveAsBin.ged> the file will be saved again as temporary hex-coded file. After this the program <hexbin> is called which again prepares the original file in the directory t:. Then this file will be copied to its original location and the temporary files will be deleted.

Before doing its changed the program <hexbin> checks the format of the hex file.
It checks whether all lines are at least 48 chars long and whether there are spaces at the necessary locations. If this check is ok. the datas will be changed.

During the check only the hex area will be observed.

During search process with the programs <searchascii.ged> and <searchbytes.ged> the first byte or char will be searched first (until a place in the file in the correct hex/ASCII area is found). Then the following chars or bytes will be compared with the remaining search string. When they are equal, the search ends successfully. If they are not equal, then the first byte or char will be searched again. This means that the search process is slower when the first byte or char is often in the file.

When inserting a byte with the program <insertbyte.ged> the actual line will be splitted and a byte will be inserted. Each last byte of a line will be inserted as start of the next line, and so on. This means, inserting a byte is slower the more lines have to be worked with after the byte in the file.

When deleting a byte with the program <deletebyte.ged> the first byte from the last line will be cut. This will be added to the line before, from which again the first byte will be cut. This will be done up to the actual line. This means, deleting a byte is slower the more lines have to be worked with after the byte in the file.

With the program <hexchar.ged> all entered chars, which are not suppressed by the editor GoldED, will be processed. For a faster speed this program is copied to t:

1.14 Known bugs

Known bugs and disadvantages:

When the edit-window will be closed by the close-gadget, the active text will be closed without warning, even if it was changed.

As the pograms is based on Arexx-macros, the operation speed is slow, especially at Amigas with no turbo cards.
This mainly is the case at search, replace, insert and delete of bytes operations.

1.15 History

History:

Version 0.8: first version, upload to Aminet

Version 0.9: second version, upload to Aminet
The following has been changed:
- now there can be spaces in the filename
- for a better overview in the hex area, spaces between long words were added
- search in hex and ASCII area was added
- the Arexx scripts were packed with the program Rexxopt of Ulrich Sibiller
- small changes
Thank you very much to Mario Kemper for his bug reports!

Version 1.0: third version, upload to Aminet

The following things have been changed:
- the macro hexchar.ged for processing the key inputs will be copied to t:, for a faster access time (t: normally is in RAM)
- removed a bug in the macro hexchar.ged at displaying in the ASCII area
- changed and extended menu
- all keys used and/or not usefull keys blocked (hopefully)
- the hex-file is write-protected now. Therefore when pushing the close gadget you are not requested to save (tip by Dietmar Eilert)
- syntay parser added (received from Dietmar Eilert) (thanks to Dietmar Eilert for his support)
- new functions added (save as, goto byte, seach with wildcard, replace, insert and delete Byte)
- extended the guide

In the moment, I have no plans for a new version.
This can change, if
- I find a bigger bug in the program
- I receive (realizable) suggestions or bug reports
- a new version of GoldED is released

1.16 Copyright

Copyright:

This program is freeware. It may be distributed free, as long as all files are included without changes. Distribution of the program at shareware CDs and shareware discs is allowed. A further commercial use of the program without my allowance is not permitted.

You use the program at your own risc. As the author of the program I am not responsible for any damage caused by use of the program. Consider, that bugs in the program are always possible.

If you want any improvements or have bug reports you are encouraged to sent it to me.

1.17 Author

Name and address of the author:

Wolfgang Morgeneier
Kurfuerstenstr. 22
D-92637 Weiden
Germany

EMail: Wolfgang.Morgeneier@t-online.de