BlockEd

COLLABORATORS

	TITLE : BlockEd		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		February 6, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

BlockEd

1.1 BlockEd V1.4 Documentation

```
              BlockEd V1.4 Oct 1993 Copyright (C) 1989-93
 Andrew Kemmis - All Rights Reserved


 TABLE OF CONTENTS


              Copyright

              Introduction

              Starting BlockEd

              BlockEd Commands

              Problems

              Things To Do

              Program History

              Compiling BlockEd

              Contacting the Author
```

1.2 Copyright

```
              BlockEd V1.4 Oct 1993 Copyright (C) 1989-93
  Andrew Kemmis - All Rights Reserved
```

BlockEd is supplied free of charge for personal use only. Commercial use
or commercial distribution is prohibited without the direct consent of
the author. Non-commercial distribution must include all files listed
below and they must not be modified. Appropriate *.info files may be
added and/or modified to suit the distribution.

The files in this distribution are:

```
  BlockEd  - executable
  BlockEd.guide - program documentation (this file)
  BlockEd.c - main source code
  BlockEd.qi - qualifier parser output (helps reading C source)
  makefile  - Aztec 'C' 3.6 makefile
```

BlockEd comes with NO WARRANTIES WHATSOEVER. The author is not responsible
for any loss or damage arising from the use of this program; the user takes
all such responsibility.

The source provided is also Copyright (C) 1989-93 Andrew Kemmis - All Rights
Reserved. It is provided to the user purely and for no other purpose than
to show how some of the program was written.

In a non-commercial distribution no charge may be made for BlockEd, other
than a nominal copy fee.

However, permission is expressly granted to Fred Fish to distribute BlockEd
on his cool collection of disks or his own CD collection disks.

Although BlockEd is free to non-commercial users, any donations would
be gladly accepted, preferably programs that you yourself have written and
find useful. See
                Contacting the Author
                for more information.

1.3   Introduction

BlockEd is a device level disk partition block editor.

It should allow you to visually edit any 512 byte block structured virtual
or physical disk partition i.e. it should work with any device that allows
you to do any random CMD_READ/CMD_WRITE of 512 bytes at multiples of 512
byte Offsets - this includes devices like PC MFM floppy disk devices as well.

BlockEd takes no specific notice of the FileSystem running on the device.

I actually first wrote this program because I didn't like any of the similar
programs around at the time. Hopefully you too will find it useful.

Currently it can only be run from the CLI because it needs a parameter to
specify the device level name of the partition e.g. DF0: (and it doesn't
work with WorkBench specified parameters with a Shift click)

1.4   Starting BlockEd

You can run BlockEd on any disk (or partition) simply by typing:

```
  BlockEd disk:
```

However, disk must be the Dos internal name for the partition
e.g. DF0: may be valid but Workbench1.3: is not if that is DF0:'s name.

BlockEd has a number of options available at runtime. The options do not
work in the standard way as yet (it's just a library function – that can
and should be changed in the future) You specify options by typing at
least the unique letters for that option (shown in uppercase further down)
after a '/' and if it is a value option using an '=' followed by the value
e.g. to run blocked with a different font you would type any of the
following:

```
 blocked /Fname=other.font/fsIZE=7 hd0:
 blocked /f=other.font hd0:
 blocked /f=other.font /fs=7 hd0:
```

This does bring about the following problem (feature) you cannot do:

```
 blocked /d0: (if the device name starts with a '/')
```

The following is the output if you specify an invalid command line:

```
usAge: blocked /Qualifiers Device
# Qual       Def      Min Max
a Quiet      false
b Datapen    1     0    3
c Selectpen  3     0    3
d Backpen    0     0    3
e Otherpen   2     0    3
f C0         0x46    0x0 0xfff
g C1         0x0     0x0 0xfff
h C2         0x9     0x0 0xfff
i C3         0xbb    0x0 0xfff
j Fname      "topaz.font" 1   25
k FSize      8     1    19
l Nullchar   '.'
m Hex        true
n Leftjust   false
o BYtemove   false
p Inhibit    false
```

Def is the default value
Min & Max are the minimum and maximum for numbers

Numbers can be entered in Hex, (with a leading '0x'), Octal
(with a leading '0') or Decimal

The options are as follows:

QUIET
  if set true don't display startup message at runtime.
  /QUIET or /!QUIET

DATAPEN
SELECTPEN
BACKPEN
OTHERPEN

```
   the gfx pens to use to draw the screen e.g. /DATAPEN=1
```

C0
C1
C2
C3
```
   the 12 bit colours to use for the Hires-Interlace Screen
   e.g. /C2=9
```

FNAME
```
   font to use (use the full name.font) e.g. /FNAME=topaz.font
```

FSIZE
```
   font size to use. Under 3.0 (I don't know much about 2.0)
   you can rescale any font to any size you like – it's built in to
   the diskfont library. e.g. /FSIZE=8
```

NULLCHAR
```
   character to use when the current character would probably display
   nothing (see the source for how it works this out) e.g. /NULLCHAR=.
```

HEX
```
   mode to start the program up in. /HEX or /!HEX
```

LEFTJUST
```
   initial justification for Decimal numbers. /LEFTJUST or /!LEFTJUST
```

BYTEMOVE
```
   initial display mode and movement – either a byte at a time or a
   long at a time. /BYTEMOVE or /!BYTEMOVE
```

INHIBIT
```
   specifies if the partition should be inhibited while using blocked
   i.e. unavailable to any other process. /INHIBIT or /!INHIBIT
```

1.5 BlockEd Commands

```
   Once the screen comes up (assuming enough memory, fonts found,
libraries found and device found etc.) the program will attempt to read
and display the Root Block according to the standard File System. This
should of course work with any File System, it just may not be the 'Root
Block'. If it could not read the block then it will display an error and
a zeroed block. The window border normally contains a Copyright message
and a version number etc. which gets written over by such error messages.
To restore the border use the Restore menu item.

   All the commands (except to change the byte offset in the block)
are shown on the screen with a highlighted character in their name.

   To move the highlighted cursor around use the arrow keys. If you
press the SHIFT key with the arrow keys, it will jump to the top, bottom,
left or right. If you use the ALT key with the left/right arrow key, it will
jump to the previous/next non-zero Byte or Long (or not move if all zero)

   To select a different block by value either click in the Offset
```

gadget or press the 'O' key and then type in the new block offset into the
gadget and press return. Without pressing return, what you typed will be
ignored.

   The first seven commands on the top right of the screen are values
calculated from the current block or partition. Press the key matching the
highlighted letter to go to the block specified next to its name. Some of
the names have no meaning or a different meaning depending on the type of
block you are looking at. If the block value is zero, pressing the key
will have no effect. If it is invalid, an error message will be displayed.

   The Checksum command displays next to it the current value of the
checksum long-word in the current block. Pressing the 'C' key will
re-calculate the value for the checksum and change the memory copy of the
block. You will have to Update the block to actually save this change to
disk. N.B. this assumes that the checksum is at long word 5 (which is not
correct on a bitmap block – and FFS datablocks do NOT have a checksum – so
don't use it on them either)

   The next four commands (+/= Advance One, -/_ Retreat One, Goto
Current & Zero Goto) are to move around the disk. The first two allow you
to step forward and backward through the disk by pressing the plus and minus
keys (with or without a shift) The Zero Goto command takes you to block Zero.
The Goto Current command takes you to the block number specified under the
highlighted cursor: If zero, it is ignored, if invalid you will get an error
message in the window border.

   ALT-ReRead lets you re-read the current block even if it hasn't
changed. (You must also hold down either ALT key)

   ALT-Update lets you save any changes you have made back to the disk
block. (You must also hold down either ALT key) N.B. if you do not do this
before moving to any other block, any changes you have made will be lost!

   The Ascii/Number Edit command lets you edit the values in the block
directly depending upon the 'Byte/Long Move' mode. If you are in Byte Move
mode an '*' will appear next to the 'Ascii/Number Edit' command and all
following keys you type will be typed into the block data at the highlighted
cursor and advance the cursor until you press the 'ESC' key (and the '*'
will disappear) If you are in Long Move mode a gadget will appear on top of
the highlighted long and let you edit it till you press return. If what you
entered was valid (or the value was already zero), the cursor will advance.
You must press 'A' again to edit the next value.

   The Spread Current command fills the entire block with the current
highlighted Byte or Long.

   Byte/Long Move switches back and forth between Byte and Long mode.
You can tell what mode you are currently in by looking at the Ascii section
of the display. If only one byte is highlighted, you are in Byte mode. If
four byes are highlighted, you are in Long mode.

   Mode Hex/Dec switches the display between Hex and Decimal.

   Justify Left/Right switches the justification in Decimal mode.

   Directory/Filename lets you type in a string to hash to the

appropriate Long offset in the current block (assuming it was a Directory
block) If you do not press return the value will be ignored (but not lost)
You can just press return to go to the value matching the string already
typed in the gadget. You can of course activate the gadget by just clicking
on it.

   The section above the Directory/Filename command just displays what
DOS thinks the device is (the driver name has had ".device" removed from the
end if it was there)

1.6 Problems

I have not tried this program on many different device types. Below is the
list which I have tried (and all have worked)

   Standard 880K low density drives
   Standard 720K PC disks with MFM device drivers
   A Meager Mega Ram Card with WD1002-05 and hard disks (on A1000)
   A590 SCSI hard-disk devices (on A1000)
   A1200 internal IDE device

I know you cannot use this program on the RAM: device or other same devices.

I have been using this program for a few years now so I would be very
supprised if anyone found any nasty bugs in it, so it should be pretty
reliable. (Of course on the above devices only I can vouch for)

If you attempt to update a device that is write protected, and you do not
deprotect it and allow it to be written, you must diskchange the disk before
you will be able to read other areas of the disk that are not buffered in
memory by the File System (even if you use another program) This happens
(in two slightly different ways) under both 1.3 and 3.0

I probably need to add some sort of 'Cancel Update' Menu option to get
around this problem.

1.7 Things To Do

Any suggestions would be very welcome.

I have tried to keep the program 1.3 compatible (especially since my main
hard-disk is actually connected to an A1000) so don't suggest anything that
is above 1.3 that cannot be optional (e.g. screen mode requester could be
optionally available on the appropriate machines)

Obviously a few things need to be done to make the program a bit more
user-friendly

The way you use the options should be standardised (extended to allow that)

A screen-mode requester would also be useful.

Also need to tidy up and release a whole collection of disk utilities I have
written (in times of dire need and have used to solve the problems.)
Look out for a few more, hopefully in the near future.

I have absolutely no idea what would be necessary to make this program work
with disk blocks that were not 512 bytes in size (assuming that a full
Root/Directory/FileHeader/etc. block is not just 512 bytes on this type of
device spread over multiple blocks if <512 bytes or part of a block if >512)

1.8   Program History

  OK there isn't much here – but maybe I'll be able to add to it as
  time goes on :-)

 V1.4
  First public release
  (Finally added gadgets to make the program somewhat user friendly)

 V1.3 and below:
  Pre-release

1.9   Compiling BlockEd

BlockEd has been written to compile under Aztec C V3.6a, however not all
of the include files or the library objects have been provided in the
distribution thus you should not (will not) be able to compile the program
yourself from the distribution.

1.10   Contacting the Author

I can be reached with comments, suggestions, bug reports etc.
at the following address till around March 1994:

    Andrew Kemmis Smith
    2/16 Nelson Street
    Penshurst NSW 2222
    Australia

or always via my parents place:

    Andrew Kemmis Smith
    32 Kenneth Street
    Longueville NSW 2066
    Australia

or by email Dec 1993 onwards (If I accept the job :-):

    a.smith@praxa.com.au