AllowBad0.5

MIKEMASTER

COLLABORATORS

	TITLE : AllowBad0.5	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	MIKEMASTER	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AllowBad0.5

1.1 AllowBad 0.5ß © Mikolaj Calusinski 1995

AllowBad 0.5 (ßeta version)

© Mikolaj Calusinski 1995. Freeware.

User Manual

1.2 AllowBad 0.5ß: Introduction

AllowBad is freeware (see Distribution) and copyright 1995-1996 by Mikolaj Calusinski. All rights reserved.

****

IMPORTANT!

IF YOU ARE A FASCIST OR NAZI YOU ARE *NOT* ALLOWED TO USE THIS PROGRAM!!!
****

1.3 AllowBad 0.5ß: Distribution

[The following text has been derived and adjusted from 'FileMaster~3.0.guide' © Toni Wilen 1995 (I don't want to reinvent the wheel). I hope~Toni, you don't mind.]

AllowBad may be distributed freely, providing the following criteria are~met:

- None of the files in the AllowBad distribution archive may be modified~or omitted.

- No money is charged for it apart from media and small handling fee.

- AllowBad may be included in freely distributable software libraries, including the Fred Fish collection and CD-ROM distributions of the~Aminet FTP site contents.

- AllowBad may not be bundled with any commercial hardware or software product without prior written consent from the author.

- You may not reverse-engineer or modify the AllowBad executable on disk~or on memory except for compressing it.

1.4 AllowBad 0.5ß: Disclaimer

[The following text has been derived from 'FileMaster 3.0.guide' © Toni~Wilen 1995 (I don't want to reinvent the wheel). I hope Toni, you don't~mind.]

THIS PRODUCT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. ALL RISKS AND~DAMAGES, INCIDENTAL OR OTHERWISE, ARISING FROM THE USE, MISUSE, OR INABILITY~TO USE THIS PROGRAM ARE ENTIRELY THE RESPONSIBILITY OF THE USER. THE AUTHOR~DOES NOT MAKE ANY GUARANTEES OR REPRESENTATIONS REGARDING THE CORRECTNESS,~RELIABILITY, ACCURACY, CURRENTNESS, ETC. OF THIS PROGRAM. THE AUTHOR WILL~NOT ACCEPT RESPONSIBILITY FOR ANY DAMAGE OR LOSSES RESULTING FROM THE USE,~MISUSE, OR INABILITY TO USE THIS PRODUCT.

1.5 AllowBad 0.5ß: What is that for?

This program was written as a replacement for good, but old ↩ BFormat 4.0.
Its~purpose is to format disks of any *floppy-based* devices, which have hard~(media) error on them. Such disks cannot be formatted (and hence utilized)~by system 'Format', so only thing you could do was getting rid of them.~Optionally, you could use BFormat of course, but such prepared disks were~very unstable under AmigaDOS, and gave 'read-write' errors quite often.

This~was because of way BFormat works - it allocates only blocks which appeared~to be corrupted, leaving rest of the track for use by filesystem. It sounds~quite logical and give you more free space on disk but doesn't work that~well with floppies. For safety of stored data it is better to avoid whole~track regardless whether there is only one or more bad blocks on it. That is~the way AllowBad handles the errors - it masks the tracks, not blocks.

AllowBad should be able to format disks in all floppy-like devices, which~have the following parameters (and these are checked when device type is~determined to be floppy or not):

- number of surfaces must be 2

- starting cylinder (LowCyl) *must* be 0

- ending cylinder (HighCyl) must not be higher than 81

- block per track can be any number above three (so at least four)

- block size must be standard (512 bytes)

- device must use AmigaDOS filesystem (0x444F53XX - 'DOS')

- max unit number is 3 (ranging from 0 to 3)

The above should rule out all harddisks (if you want to format damaged~harddisk use BFormat or Quarterback), CDROMs, etc. AllowBad works (and has~been tested) with trackdisk (DFx:) and diskspare [(c) 1992-1994 by Klaus Deppisch] (DSx:) devices. It should~support all possible future devices which comply to the specs mentioned.

The program is aware of high density (HD) floppies and should support them,~but unfortunately I was unable to test it (hence the ßeta status of~AllowBad). Please, report all the possible bugs to the
address
found at
the~end of this document.

You can format RAD: device with AllowBad (as long as it complies to above~specs) but this does not make much sense.

1.6 AllowBad 0.5ß: Requirements

As most of nowadays programs, AllowBad requires at least ↩
version 2.04
of~operating system. Reasons are obvious: it is much easier for programmer (me)~to code the whole thing and AllowBad works only with FFS types of disk (1.3~doesn't support FFS floppies directly). If you want to use DIRCACHE~filesystem you need OS 3.0 or higher (however AllowBad can format dircache~disks under 2.04, too).

AllowBad also requires some free memory; its amount depends on device. For~further details see
Way of operation

section.

1.7 AllowBad 0.5ß: Limitations

AllowBad is somehow a little bit more tollerant than other programs of its~kind - it can properly format and initialize disks, which have tracks 0~and/or root damaged! The only requirement for such disks is the ability to~read/write first two block on track 0 (ie. bootblocks) and one block on~middle track of disk (ie. rootblock). So, if you have for example the disk~which has damaged only block number 3, there is a good chance AllowBad will~be able to initialize this disk and make it available for file storage (with~low side of track 0 allocated as bad). But remember: such a disk is~extremely vulnerable and unstable. Be sure not to store any important files~on it!

This version of AllowBad is meant as a CLI command and cannot be used from~Workbench (it has no GUI, so no pain I think).

1.8 AllowBad 0.5ß: Usage

Standard template (can be obtained via use of question mark) looks like~this:

AllowBad DRIVE/K/A,NAME/K/A,INTL=INTERNATIONAL/S,DIRCACHE/S:

As you can see only two parameters must be specified - drive name and name~you want your disk to have. Drive name is standard doslevel device name such~as 'df0:', 'ds1:', etc., case insensitive. Must end with colon. Name of the~disk must not exceeded 30 chars (this limit is imposed by current version of~AmigaDOS). Names longer than permitted will simply be truncated. If you want~the filename to contain spaces, you must use quotes. Remaining two~parameters are optional and pertain to type of filesystem used while~initializing the disk. INTL (equivalent to INTERNATIONAL) denotes INTL FFS~(DOS3) and DIRCACHE - DC FFS (DOS5), respectively. If you use both of them~at the same time DIRCACHE will be used (DC implies INTL anyway). When none~of switches is present the disk will be formatted with default normal FFS~(DOS1).

For some technical reasons OFS types of filesystem are not supported. I hope~you can live with it however.

While formatting, the program can be interrupted with CTRL-c. If this is so,~execution is terminated with RC (return code) set to 5 (WARN) and mesage~'***BREAK' is printed.

1.9 AllowBad 0.5ß: Way of Operation

At start AllowBad checks whether the parameters issued by user are correct~(they are obtained through standard ReadArgs() function). At this analizing~stage, availability of disk unit is also confirmed. If disk is not in use,~AllowBad immediately allocates it (by Inhibit() function). Else

program~quits with error message 'AllowBad failure: object is in use'. Unfortunately~some programs (such as great disk editor DPU 1.5 or recovery utility~Quarterback Tools 2.2) ignore the fact that the drive is inhibited by~someone else and allow you to work with such a drive. This could lead to~interference between those programs and AllowBad, resulting in incorrectly~formatted/initialized disks. This is NOT my fault, but programmers of those~utilities! My advice is not to use any disk utility while AllowBad~executes.

If everything went okay (and there is a write-enabled disk in drive) two~memory buffers are allocated (and these are the only memory allocations made~by the program) - first buffer is the three times of one track in size~(calculated as follows: 3*BlocksPerTrack*512), and the second one is 83~bytes. Thus currently the most 'memory-hungry' situation I can think of will~be HD disk formatted using diskspare.device [(c) 1992-1994 by Klaus Deppisch]. In this~case AllowBad will need 3*24*512+84 (= 36948) bytes of free memory. The type~of allocated memory (chip, public, etc.) depends on flags parameter in~device's FileSysStartupMsg (you can specify it in BufMemType parameter in mountlist).

Because Allowbad is system friendly, it uses standard device calls for~accessing the disk (via IORequest). First, each track of disk is formatted~(using command #11 - FORMAT) with special data pattern. Then (after flushing~device buffers for reliability) contents of this track is read back from~disk and compared with pattern data. If operation is successful, the same~track is written with zeros and once again verified. During all of this~program informs briefly about what is going on. Each error encountered is~notified in the second buffer and the appropriate message is diplayed. Each~track can be accessed up to four times, so AllowBad undoubtedly is not the~fastest formatter around. But when dealing with corrupted media, reliability~is what counts, not time.

When formatting completes, AllowBad attempts to initialize the disk - first~boot blocks are written (and verified) then so is root. If above could be~done, disk state info (as obtained during format stage) is analyzed and~place for other control blocks is determined. These blocks include: - dircache~info block (if DIRCACHE switch was specified), bitmap block, file info block~and eventually some file extension blocks. The last two can be needed only~if there were errors detected.

AllowBad masks corrupted areas using dummy file which pretends to occupy all~the bad tracks. This keeps the AmigaDOS filesystems from using those places~and you can store files the same way as on good (error free) disks. To avoid~confusion the file is protected from reading, updating, deleting, etc. and~should remain as such (don't play with this file - it IS completely faked~and does not contain anything intreresting!) The name of the file is~'dummy.bad'.

Because of way AllowBad works (and I can't think of anything better, can~you?) disks prepared by it can *only* be used under DOS, even that with some~limitations. Don't try to diskcopy to such a disk! Use DOS command 'copy'~or any file managing program instead. Also, don't try to optimize bad disks~with Reorg or like. If you want fast directory listings use dircache (OS 3.0~rules!) And remember that AllowBad DOES NOT repair anything - it only allows~bad disks to be used while they actually still remain bad!

On completion AllowBad informs you about number of blocks allocated (this~not

includes boot, root, DC, bitmap and FIB blocks).

1.10 AllowBad 0.5ß: Bugs

During my tests (well, not very intensive) no bugs were detected, which does~not mean there aren't any. If you find any error or have some ideas of~improvement (or if you simply wanna chat a little, receive the source, etc.)~please contact me!

1.11 AllowBad 0.5ß: Contact address

Unfortunately, I still have no direct access to Internet (must buy a modem~first) so you can reach me only by snail mail, sorry. Here is my address:

Mikolaj Calusinski
ul. Olsztynska 113/117
42-200 Czestochowa
POLAND

I hope you find this little proggy useful. Have fun!

Mike.

----------> AMIGA - THE BEST COMPUTER EVER <----------