

CompressDisk

COLLABORATORS

	<i>TITLE :</i> CompressDisk		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 16, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	CompressDisk	1
1.1	main	1
1.2	Introduction to CompressDisk	1
1.3	What is CompressDisk	2
1.4	Installing CompressDisk	2
1.5	Using CompressDisk	3
1.6	CompressDisk compression algorithm	3
1.7	The LZRW3a algorithm	3
1.8	The Standard Calgary Corpus	3
1.9	Performances of LZRW3 and LZRW3A algorithms	4
1.10	Technical informations	4
1.11	Configuring CompressDisk	5
1.12	CompressDisk performance	5
1.13	Recovering from errors	6
1.14	The advantages of using CompressDisk	7
1.15	Acknowledgements	7
1.16	Copyright informations	7
1.17	Keeping the developers developing	7

Chapter 1

CompressDisk

1.1 main

CompressDisk

This is the documentation for CompressDisk V1.9B, a disk doubling package for AmigaDOS.

Copyright © 1992,1995 Antonio Schifano.

[Introduction](#)

[What is CompressDisk?](#)

[Installation and upgrading](#)

[Configuring](#)

[Guidelines](#)

[Error recovery](#)

[Compression algorithm](#)

[Compressdisk and other disk doublers](#)

[Performance](#)

[Technical info](#)

[Copyright - distribution](#)

[Acknowledgements](#)

[How to register](#)

1.2 Introduction to CompressDisk

Data compression is not a new thing in the Amiga scene: there exist many commercial and public domain utilities that give to all Amiga users the possibility to save disk space.

We can roughly divide data compressors in two main classes: the class of archivers (like ZOO, LHARC, etc.), and the class of "file oriented" compressors (like COMPRESS, POWERPACKER, IMPLODER, etc.). Archivers' main feature is their ability to preserve the files hierarchy; the price to pay is the impossibility to use the archived files in the compressed form.

On the contrary, file oriented compressors sacrifice this feature to achieve a better usability of compressed files: programs compressed by PowerPacker, for instance, remain executable; moreover, the package includes some useful tools (like PPMORE) that compensate for the versatility lost because of the compression (for instance, you can still read compressed text files, but you cannot edit them).

CompressDisk tries to resolve the limitations of both types of packers by a different approach to the problem.

1.3 What is CompressDisk

CompressDisk is a disk doubling package based on an Exec device.

It allows you to create virtual partitions on your hard-disk which are usable like normal partitions, but with the great advantage that every file that you write in them is automatically compressed (and obviously, it is decompressed whenever you read it back).

Files stored on virtual partitions can be rewritten (even partially) and applications can even Seek() into them as with usual files.

1.4 Installing CompressDisk

The device can be installed only on Amigas with kickstart V37 or better.

The installation is (almost) totally performed by the programs "CDInstall" and "CDPartition", that take care of copying all necessary files in the right directories:

- The file "compressdisk.device" is copied in the DEVS: directory.
- For each virtual partition desired, you can create an oportune entry in the DEVS:mountlist file with CDPartition. If you are using Workbench V38+, all entries will be created in the DEVS:DOSDrivers directory.
- The preference editor "CDPrefs" is copied in the SYS:Prefs directory.

Notice that CDInstall and CDPartition need a standard environment to work properly; I received many complains from people who claimed that these programs refuse to work on their machines, or have strange behaviours: anyway, CDPartition and CDPrefs have been tested on many Amigas (including the A4000) and under OS2.04 (even with Workbench 2.1) and 3.0. Make sure, in particular, that the environment variables "Kickstart" and "Workbench" exist.

For the device to work properly, you should add the following line to your s:startup-sequence file, just after the command SetPatch:

```
assign CDISK: <directory>
```

where <directory> is the complete path of the directory in which you want the device to store all the compressed tracks of every unit used.

Since version 1.8, it is possible to specify a directory for each unit: simply assign the names CDISK0, CDISK1, CDISK2 and CDISK3 instead.

For instance, if you specify

```
assign CDISK1: hd0:unit1
```

```
assign CDISK: hd1:tracks
```

the device will store the tracks of unit 1 in the directory "hd0:unit1", and the tracks of units 0, 2 and 3 in the directory "hd1:tracks".

If neither CDISK<n>: nor CDISK: are specified, the device will put the tracks in the root directory of the boot partition.

Since version 1.9 the device does not store all the tracks in the same directory; it creates several subdirs and distributes the tracks into them.

This translates into a *faster* access to the tracks, and avoids the performance loss that afflicted previous versions of compress-disk, where the time to load the tracks could get much longer than the time to decompress them (and even worse, was directly proportional to the number of tracks).

As always, there is a price to pay: the subdirs cause an overhead of 512 bytes per 2M of data (hope you can bear it...)

This speed gain is particularly evident when you use ZeroUno or a similar utility to clear the empty sectors of a virtual partition: you only need a fraction of the time you had to wait before!

Once mounted, each virtual partition must be initialized; this can be easily accomplished with the system command "format". Since the device stores only the tracks written at least once, it is advisable to use the QUICK option: this will let you save disk space and time (non-quick

formatting of a 40M partition can take more than thirty minutes on a standard Amiga 500).

Important: if you are upgrading from pre-1.9 releases of compressdisk, the "CDInstall" script will execute the the supplied "Convert" command in order to distribute all the track files into the appropriate set of subdirectories needed by the device. Please make sure you have at least 25K on CDISKn: for each 100M partition you use, or the conversion could fail.

When the installation is completed you must reset your Amiga as soon as possible, since the tracks reorganization is not compatible with the pre 1.9 releases. The version already loaded in memory will thus no longer be able to load the tracks, and in this situation weird things may happen.

Be careful!

1.5 Using CompressDisk

You can use virtual partitions created with CompressDisk like usual ones; nevertheless, it is better to follow some guidelines.

First of all don't write any files into any virtual partition of any unit whose work disk is full: if the device cannot store some tracks, you may lose some data.

Moreover, you should not use virtual partitions for storing temporary files; remember that deleting a file (on the standard file systems), does not clear the blocks that it occupied (in fact, you can recover it using proper utilities), until the file system overwrites them. This means that when you delete a file you will NOT free any space on the unit work disk.

Virtual partitions created with the partitioning program use the Fast File System; however you can use any file system you want. Since the Standard File System and its international and directory-cached versions reserve part of the data blocks for internal informations, they can reduce the tracks compression; for this reason their use is not recommended.

Kickstart V39+ users are advised to use the new directory-caching file systems.

1.6 CompressDisk compression algorithm

The most decisive choice in the device realization was the selection of the compression algorithm: it is clear that the device performance depends on the data compressor and decompressor. The algorithm had to be as fast as possible, had to use little memory, and had to be reasonably good at compressing.

The algorithm used by CompressDisk is LZRW3, derived from Ross Williams' **LZRW3-A**, which features an average compression speed of 20K/sec, an average decompression speed of 75K/sec and needs only 32K of memory.

See [table 1](#) for a rough idea of the performances obtained by the LZRW3-A and the LZRW3 algorithms on the **standard Calgary corpus** files.

CompressDisk release 1.9 is backward compatible with all previous releases (old partitions will work fine under the new version), provided that the '**Convert**' utility has been run.

1.7 The LZRW3a algorithm

For further informations about LZRW3-A refer to the original documentation given by Ross Williams (directory "pub/compression" on the FTP site [sirius.itd.adelaide.edu.au](ftp://sirius.itd.adelaide.edu.au) [IP=129.127.40.3]).

1.8 The Standard Calgary Corpus

The "standard Calgary corpus" is a set of files used to evaluate the practical performance of compression algorithms. It can be found in the directory "/pub/text.compression.corpus" on the FTP site [fsa.cpsc.ucalgary.ca](ftp://fsa.cpsc.ucalgary.ca) [IP=136.159.2.1].

1.9 Performances of LZRWS3 and LZRW3A algorithms

```

-----+-----+-----
| lzrw3a | lzrws3
-----+-----+-----
file size | size K/s K/s % rem | size K/s K/s % rem
| com dec | com dec
||
bib 111261 | 49047 17.8 70.9 44.1 | 45268 20.1 77.3 40.7
book1 768771 | 416139 15.3 65.3 54.1 | 390213 16.9 69.5 50.8
book2 610856 | 274378 18.3 74.8 44.9 | 260781 20.0 79.4 42.7
geo 102400 | 84221 8.7 34.0 82.3 | 81333 9.3 35.6 79.4
news 377109 | 191298 15.2 60.3 50.7 | 174211 17.3 66.7 46.2
obj1 21504 | 12655 11.9 44.8 58.8 | 12388 11.9 44.8 57.6
obj2 246814 | 108048 17.1 68.5 43.8 | 101917 18.5 68.4 41.3
paper1 53161 | 24530 17.0 63.3 46.1 | 24170 17.7 72.8 45.5
paper2 82199 | 39486 17.1 68.5 48.0 | 38469 18.5 76.1 46.8
pic 513216 | 111008 40.7 133.3 21.6 | 72802 38.8 137.6 14.2
progc 39611 | 17926 17.3 66.0 45.3 | 17482 17.4 66.0 44.1
progl 71646 | 24366 23.8 85.3 34.0 | 22232 25.9 85.3 31.0
progp 49379 | 16808 22.9 82.3 34.0 | 15136 27.4 82.3 30.6
trans 93695 | 30292 23.7 86.7 32.3 | 25122 29.9 97.6 29.9
-----+-----+-----
average 224402 | 19.0 71.7 45.7 | 20.7 75.7 42.9
total 3141622 | 1400202 | 1281524

```

TABLE 1 - LZRW3A and LZRWS3 performances on the standard Calgary corpus (A500 with MC68000 at 7.16MHz).

1.10 Technical informations

The device core is a process which satisfies all I/O requests: whenever it receives a CMD_READ command, it loads and decompresses all the needed tracks; in a similar way, whenever it receives a CMD_WRITE or CMD_FORMAT command, it compresses and stores the tracks.

From a technical viewpoint, compressdisk.device may look a bit strange, since it seems to break the hierarchy of the Amiga OS. In fact it is unusual for a device to use the DOS, since it is at a higher level in the hierarchy. Yet the hierarchy is purely functional: devices are used for management of physical resources (eventually virtual ones) and handlers for O.S. logical resources; for this reason a device is allowed to implement its primitives using a logical resource, provided that the tasks which use the device are not aware of the implementation of the primitives.

To speed up read/write operations, the device features a built-in cache in which it stores the most recently used tracks. The cache size is user configurable.

If possible, the process uses the cache both in reading and writing: in reading, if a track is already in the cache, there is no need to reload and decompress it; in writing, tracks are compressed and stored only if there is no space left in the cache.

Tracks are updated upon receiving a CMD_UPDATE command; data consistency is thus ensured.

1.11 Configuring CompressDisk

To choose your CompressDisk configuration, execute the "CDPrefs" tool.

You can choose how many track buffers must be used by each unit, and a global updating strategy for them.

The track buffers can range from 1 to 32. Every track has 64 sectors of 512 bytes each, for a total of 32K. The device needs about 75K of working memory, plus about 32K for each track buffer. Notice that no extra memory is needed by the device for opening the files of a virtual partition.

The default configuration is eight tracks per unit; many tests showed that the device is usable with four tracks or (if you want to save your memory) even with two tracks, whereas with only one track the performance reduced significantly, especially in writing.

For the updating strategy, you have three possible choices:

- Standard: buffers are updated as soon as the device receives a `CMD_UPDATE` command. This is the standard behaviour of Amiga devices.
- Smart: the device updates the buffers only if a `CMD_UPDATE` command has been received and a certain (user-programmable) amount of time has elapsed without receiving other write requests.
- Safe: in this mode the device works in the same way as in standard mode; the device, however, tries to preserve the virtual partitions consistency by making an uncompressed copy of the cache buffers to disk for each `CMD_WRITE` command. The compression of tracks is performed only when they must be discarded from the buffers, or when a `CMD_UPDATE` command is received. In version 1.5 tracks used to be compressed each time they were written to disk, with great loss of performance (indeed, I think it was almost unusable). The only difference is that if a software failure happens while writing to a virtual partition, you will end up with some uncompressed - yet still readable! - tracks; however, they will be compressed as soon as the device reads them.

The best performance is obtained in SMART mode, that avoids all overheads caused by the repeated `CMD_UPDATE` requests issued by the standard AmigaDOS file systems.

The default timeout the device uses in SMART mode is 5 seconds; however, it can range from 1 to 60 seconds. If a software or hardware failure happens before the device flushes its buffers, you may lose some data; it is therefore advisable to use short timeouts (at least if you don't know what you are doing).

The configuration can be changed at run time.

1.12 CompressDisk performance

The performance of CompressDisk is greatly influenced by a lot of parameters:

- ? the Amiga model (i.e., the processor type and its speed);
- ? the hard disk speed and type (DMA or not);
- ? the **configuration** you have chosen;
- ? if you registered or not :-).

I could not use any disk speed timing programs, because those I own got confused by the caches, giving RAD-like unattendable results. Instead, I measured the time taken to copy some files to a virtual partition.

The following table shows the time taken to copy the directory containing the **standard Calgary corpus** (about 3,1M) from RAM: to a virtual partition; timings refer to the registered 68020 version of CompressDisk, using 8 buffers of cache, on a standard A1200 with fast RAM and a 360K/sec 40M drive:

Mode Time Speed

Smart (5s) 1m 45s 30.28K/s

Standard 1m 53s 28.13K/s

Safe 3m 15s 16.30K/s

Since the **standard Calgary corpus** contains mostly big files, there is little DOS overhead, thus the device performance in "smart" mode is the same as in "standard" mode; for operations with shorter files, however, operations in "smart" mode are considerably faster; the following table reports the time taken to copy the assembler include files v3.1 (about 631K) to a virtual partition (with 4 buffers):

Mode Time Speed

Smart (5s) 0m 22s 28.68K/s

Standard 0m 30s 21.03K/s

Safe 1m 50s 5.73K/s

The highest cylinder number currently supported is 65000. The largest possible size of a virtual partition is thus 2031.25M.

The evaluation version supplied in this archive is about 35% slower.

Registered users will receive the optimized 68000 or 68020+ version of the device, according to their will.

1.13 Recovering from errors

Whenever CompressDisk meets an error condition, it will inform you (via a requester), thus giving you the possibility of recovering.

There are actually two possible error conditions:

i - Compressdisk is unable to write - for any reasons - a track buffer to disk.

In this case, a requester will inform you about which track of which unit could not be written. By hitting the proper gadget in the requester you can then decide to:

RETRY the write operation, or

IGNORE the error condition.

For instance, if the work disk of a unit gets full while writing to a virtual partition, a requester will inform you about this; you can make some space on the disk (by deleting some files) and then retry, or ignore the error (in this case some data will NOT be written to disk - beware!).

ii - Compressdisk detects an error while decrunching a track. This can be caused for instance by viruses trashing your hard disk. Again, a requester will inform you about this, and you can decide to:

RETRY: the device will again try to read the track, and decrunch it. If the error was a transient read error from your hard disk, this choice will lead to a 100% error recovery.

IGNORE: simply the device ignores the error condition, and you will get the data as if it had decrunched OK (with the only difference that it will probably be garbage).

FIX: like IGNORE, with the difference that the track is fixed, i.e. the decrunched data/garbage is written back to disk. This will prevent the device from signaling the error condition each time you read that faulty track.

FAIL: acts as a standard Exec device, i.e. signals to the reader that the requested track cannot be read.

If you ever get this requester, RETRY at least once (never knows...).

1.14 The advantages of using CompressDisk

Someone has expressed some diffidence about using CompressDisk, and mainly because you cannot read the data stored in a virtual partition without it - apart for the set of tracks, I mean.

There is however a good reason for using these tracks: simply, this frees compressdisk of most of the problems that afflict similar packages:

- CompressDisk needs no extra memory for opening the files of a virtual partition;
- CompressDisk automatically joins small files together, obtaining a much better compression than with a separate packing;
- CompressDisk packs also file headers, file extension blocks, and all the informations required by the file systems;
- CompressDisk has proved to be the most stable and compatible among the disk-doublers.

Moreover, data security is not invalidated. In fact, if a virtual disk becomes inconsistent for some reasons, you can fix it or recover most of the files by using standard tools, like DiskSalv.

1.15 Acknowledgements

I wish to thank the following people, in no particular order:

- Raffaele Dell'Aversana, for the academic discussions prior to the realization of the device;
- Ross N. Williams, for releasing the LZRW* material;
- Jan van den Baard, for the (useful!) utility "GadToolsBox";
- Giuseppe Sacco, for the utility "ZeroUno" (see the DISCLAIMER!! ;-);
- my sister Letizia, for no particular reason;
- all registered users;
- Gaetano Di Caprio, for some hints in the preparation of this tutorial.

1.16 Copyright informations

CompressDisk, Copyright © 1992, 1995 Antonio Schifano.

All rights reserved. This program is SHAREWARE: if you use it, please [register](#) .

The program can be distributed only in its original archive format; can be included in collection disks of public domain software. It can also be distributed in electronic form via BBS, FTP, and similar. You may not charge more than Fred Fish does for a single library disk. Also no profit is to be made by selling this program. The commercial use and distribution of this program is NOT allowed without written permission from the author.

The author cannot be held liable for any damage or data loss directly or indirectly caused by the use or misuse of this program; by deciding to use it, the user assumes all the responsibilities.

1.17 Keeping the developers developing

To register, please compile the registration form included in this archive and send it, together with Lit. 15000 if you live in Italy, 20DM if you live in the European Community, or 20USD if you live somewhere else, to the address below.

Registered users will receive the registered version, and will be able to upgrade to future releases of CompressDisk at the cost of disk, envelope and postage (Lit. 5000, 5DM or 4USD as above, or free if you have an Internet address).

Send cash, hints, registration forms to Antonio Schifano, at the address:

via dell'Oca, 11

91100 Trapani, Italy

INTERNET: schifano@cli.di.unipi.it

FIDO: 2:332/620.10

Thanks.
