DiskSpare

COLLABORATORS

	TITLE : DiskSpare		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		November 23, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

DiskSpare

1.1 DiskSpare.guide

DiskSpare.Device (DSD) V 3.0 copyright 1992-1994 by Klaus Deppisch

BriefInfo   Hints   Information

Installation   Flags   Programmers

Mountlist generally   Changes   @{ " V36NoClick " Link "Support/V36NoClick. ←
    doc/main"}

Mountlist and DiskSpare   Problems   @{ " QuietTD " Link "Support/QuietTD.doc ←
    /main"}

Device generally   Address   PFS

DiskSpare especially   Distribution

1.2 copyright

```
Distribution Conditions
~~~~~~~~~~~~~~~~~~~~~~~~~
  The device is FREEWARE. This means, you may freely copy it
  as long as you don't modify it (in ANY way) and as long as you
  distribute ALL the files in this package with it.

  You may use it with commercial or shareware programs
  if you say CLEARLY that this device is FREEWARE !

  You MUSTN'T reassemble the device itself nor modify
  this doc.file or any other file in this package.

  If you need a modified version please contact me.
  (see also support/diskspare.doc)

  This disk-format MUSTN'T be used for stupid NDOS-Games !!!
```

```
 MOST IMPORTANT:
  You use this software at your OWN RISK.

  Under NO CIRCUMSTANCES will the author be liable for any
  direct or indirect damage or data loss resulting from
  the use or misuse of this software or the documents.

  All rights are reserved by the author.
  Suggestions and (detailed) bug-reports are welcome.
```

1.3 briefinfo

```
BriefInfo
~~~~~~~~~
  DiskSpare is a completion or replacement of the trackdisk.device.
  It  lets  you  store  more data on your disks faster than usual.
  For this a custom mfm format is used. -> No online compression !
  The capacities are:

  - 984K with DD disks (normally 880K)
  - 1968K with HD disks (normally 1760K)

  Diskspare.device requires Kickstart Version 2.0 or higher.

  If you want to test diskspare.device (DSD from now on), simply
  click on the 'QuickTest' icon. The script mounts a drive 'DS0:'
  on Unit 0 (your DF0: drive) with 984K capacity. To format the
  disk, open a shell window and type: 'format drive ds0: name test'
  <return>. If you mount any device after 'LoadWB' you need the
  shell to format a disk. If you mount DSD during boot, you can
  use the Workbench format as for DFX. After formatting, you should
  be able to use the disk like any other disks.
```

1.4 installation

```
Installation
~~~~~~~~~~~~~
  The easiest way to install diskspare.device (DSD) on your hard
  drive or boot-disk is to use the included install script. The
  only thing you have to do, is to click on the 'English' icon.
  The script asks you which drives you want to install and if it's
  a HD drive or not.
  It also notes which Workbench version you're using. There's one
  thing to note: The script needs some programs of your C: dir. So
  if you're using a boot-disk which doesn't have the required progs
  in its C: dir it will fail.

  But a 'hand-made' installation is no problem, either.

  - copy the file 'diskspare.device' into your Devs: dir(ectory).

  - If you use Workbench 2.1 or higher, you only have to copy the
```

desired Mountfile(s) from the 'DD' or 'HD' drawer into your
'Devs:DOSDrivers' dir. The drive will automatically be mounted
after the next reboot.

- If you use Workbench 2.0, you have to take some more steps.
  Append the file 'Mountlist' to your personal Mountlist (to be
  found in the 'Devs:' dir using a text editor.

  WARNING: NEVER copy ANY mountlist to your devs: dir. This would
  overwrite your personal mountlist. Losing your mountlist may
  cause a total breakdown of your system !!!

  After appending the mountlist you have to add two lines to your
  's:User-Startup'. Insert first 'mount DSX:' then 'addbuffers
  DSX: >NIL:' using a text editor (e.g. ED). The drive(s) will
  be mounted within the next reboot.

If you use the install script, drives will be installed with
984K (DD) or 1968K (HD). If you want to change this, or add the new
verifying option, you have to edit the corresponding mountfile. The
size of the disk is controlled by the 'HighCyl = ..' and by the
 Flags field.

1.5 flags

```
Flags
~~~~~
  Bit value Meaning
    when
    set

  0 1 Bit is 0: Diskchange works only on the DSx drive.
      Bit is 1: Diskchange works on both DSx and DFx.

      This Bit should only be set if you use the FFS.
      (DosType 0x444F530x DEFAULT!)

  1 2 Bit is 0: DSD uses 80 cylinder. (960K/1920K)
      Bit is 1: DSD uses 82 cylinder. (984k/1968K)

      Please note:
      If you use more than 80 cylinders, the trackdisk
      device must be patched !
      The MountList field 'HighCyl = ' must also be
      changed accordingly.

  2 4 Bit is 0: The FFS uses only the values in the
      mountlist. -> No DD/HD detection.
      Bit is 1: The FFS asks DSD for the correct
      disk size.

      Please note:
      This bit should only be set when using the FFS.
      If it's set, the FFS is patched.
```

3 8 This Bit isn't used because DSD V3.0 doesn't need
    fastram anymore.

4 16 Bit is 0: Normal writing mode
    Bit is 1: EACH trackwrite is verified.

    Please note:
    If you set this bit, you (nearly) half the writing
    speed. BUT you can be sure that the data on the
    disk is ok.
    Especially useful when using cheap disks.

If you want to change the flags, you only have to add the values
of the bits you want to set (1). Unset bits are '0'. This value
is to be inserted in the 'Flags = xx' line.

Example:

You want to use a HD drive with 82 cylinders and the diskchange
should work on DFx, too. For this you have to add 1 + 2 + 4 = 7.
The flags line is now 'Flags = 7'.

1.6 changes

Changes:
~~~~~~~~

  Version 3.0

  The doc.file changed to AmigaGuide format.

  Device changes:

  - completely rewritten

  - noticeable faster (especially on slow Amigas)

  - switchable Verify Write. A (very good) idea of Kriton Kyrimis.

  - No more fastram used. For this more chipram and the blitter.

  - The DMA buffer is allocated accordingly the inserted disks.
    No waste of memory with DD disks.

  - The HD mode (of the device!) now always works. In earlier
    versions it could be switched off totally. (nonsens)

  - again bugfixes.

  Version 2.2

  - works now with PFS of Michiel Pelt. Mountfiles inlcuded.
    The names PS0 and PS1 are from Brett Simcock.

```
  - bugfixes


  Version 2.1

  - has a false start


  Version 2.0

  - the first version of diskspare.device which works with HD
    disk. My 'highdensity.device' is now dead.


  Version 1.6

  - first public release.
```

1.7 mountlist1

```
Mountlist generally
~~~~~~~~~~~~~~~~~~~~
  The mountlist is an ASCII file which contains the information
  the system needs to use this drive. Since this is important
  information I would like to explain it.

  Device = ...  This line tells the filesystem which device it
      should use. The device must be a trackdisk
      compatible device.

    Unit = ...  The value behind the unit tells the filesystem
        which unit of the device should be used. The
        value can be 0...X. DSD and trackdisk support
        four units (unit0...unit3). Harddisk devices
        allow more units (e.g. 8)

   Flags = ...  The flags can be used to tell the device some
      specific settings. They are used intensively
      by DSD.

      Surfaces = ...  Number of heads of the device.

     BlockSize = ...  It should be called 'SectorSize' because it
          tells the filesystem the size of a sector in
          bytes. Most devices support sector sizes of
          512 bytes.

BlocksPerTrack = ...  Should be called 'SectorsPerTrack'. It tells
      the filesystem how many sectors a track contains.

      Reserved = 2  How many blocks (or sectors?) should be reserved
            for bootblocks. It MUST !!! be 2 (two) when using
            the FastFileSystem.
```

```
     Interleave = ...  Only useful for harddisk devices.

  LowCyl = ...  Tells the filesystem the lowest cylinder to use
     for a drive (partition).

      HighCyl = ...  The upper cylinder. If a disk has 80 cylinders
           (and the LowCyl = 0) it must be set to 79.
           The computer starts counting at '0'.

      Buffers = ...  This number tells the filesystem the amount
           of cache buffers it should use for this drive.
           For floppy disks it shouldn't be less than
           50 ! If you use too few buffers, the speed will
           decrease enormously.

    StackSize = ...  ALWAYS use the default value of your filesystem!

      DosType = ...  A filesystem specific flag (same as the flags
           for the device).

  Mount = 1  This flag is very important, too. If it's set,
     (1) the drive will be mounted completely. This
     means the device and filesystem are loaded and
     all allocations are completed. If it isn't set
     (0), the device and filesystem are NOT loaded
     until the drive is accessed for the first time.
     This may (in fact it does) cause problems with
     some programs.
```

1.8 mountlist2

```
Mountlist and DiskSpare
~~~~~~~~~~~~~~~~~~~~~~~~~

  The devicename must be 'Device = diskspare.device'.

  The unitnumber can be set to 0-3. If you use unit '0',
  the DF0: drive will be used. (Same numbers as for DFx)

  The  Flags  can be set to fit your needs.

  Drivegeometry:

    Surfaces     2
    SectorSize     512
    SectorsPerTrack  12 (DD) or 24 (HD)
    LowCyl       0
    HighCyl      79 or 81 (according to the Flags)

  If the autodetect bit (2) isn't set, you can change the lowcyl,
  highcyl, sectorsize and sectorspertrack. You only have to make
  sure that the disk capacity is correct.
  BUT:
  Enlarging the sector size isn't good for anything but testing
  your diskmonitor (most of them fail with other sizes than 512
```

```
bytes/sector).
If you use 1024 or more bytes/sector, you waste much disk space.
A simple (silly) example:
You use 2048 bytes/sector. If you save a file of 1 (one) byte
length, you waste 2047 bytes (for data only). If you use 512
bytes/sector, and save 1 byte you waste only 511 bytes. You see
big sectors waste much more disk capacity. Remember, DSD wasn't
written to waste capacity but to get more capacity per disk. If
you want to waste MUCH disk capacity buy a PC, where you MUST use
2048 and more bytes/block!

The other lines in the mountlist belong to the filesystem.
```

1.9 device1

```
Device generally
~~~~~~~~~~~~~~~~~
  Since many people don't know what a device does and especially
  what a device doesn't do, I would like to explain it.
  It's quite simple. A (software) device is a program, which allows
  the OS to access a specific hardware. It essentially reads and
  writes data from or to this hardware. If a device should be used
  in a filesystem, it must be able to do some additional things,
  like seeking, motor on/off and other things. Quite simple.

  A device doesn't know the data on the sectors it transfers to the
  filesystem (at least DSD doesn't). The advantage of this is, that
  you can access a device by various filesystems.

  Here's the way of the data from ram to disk:

  Application or Shell    wants to write data

  dos.library     checks which filesystem manages
          the desired drive and sends it
          the request of the application.

  Filesystem      checks which device the drive is
          on, whether there're enough free
          blocks to store the data on.
          Updates the directory, marks the
          blocks as being used and sends
          a request to the device to write
          the data at sector x to sector y.
          Simplified !

  Device        checks whether the sector numbers
          are valid and writes the data.

  That's the (simplified) way from ram to disk. As you can see, the
  device is only the errand boy of the filesystem.
```

1.10   device2

```
DiskSpare Especially
~~~~~~~~~~~~~~~~~~~~~
  DSD nearly acts like any other device. 'Nearly' because of
  the SWITCHABLE patches (trackdisk and FFS). But DSD doesn't
  know the data on the sectors. It only knows how to read/write
  them to disk.
```

1.11   which problems?

```
Problems
~~~~~~~~
  Here I want to discuss some problems. Problems which are caused
  by bugs, also see 'Support/Programmers.doc' known bugs.


  A. The Mountlist Problem
  ~~~~~~~~~~~~~~~~~~~~~~~~~
  If you've changed the mountlist, and DSD doesn't work properly
  (NDOS disks) it's quite sure that there's only an error in
  the mountlist and not in the device. See also  Flags .


  B. The Drive Problem
  ~~~~~~~~~~~~~~~~~~~~~~
  Since diskspare.device uses a custom MFM format, it's not 100%
  sure that any drive will work. Do you know anything which is
  100% sure ?  So far, nobody told me that he's got a drive which
  doesn't work.


  C. The FileSystem Problem
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~
  The problem is that DSD can change the disk capacity (DD/HD).
  Since some harddisk devices don't (or didn't) work properly,
  the FFS allows only the trackdisk.device and the carddisk
  device to change their disk capacity. To make it possible that
  DSD is also asked about the diskgeometry after a diskchange,
  the FFS must be patched. It might be possible that future
  releases of the FFS cause problems.


  D. The trackdisk.device Problem
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  There's only a problem if you use more than 80 cylinders. Default
  is 82 ! The trackdisk.device itself handles only 80 cylinders.
  -> It must be patched to use 82 instead of 80 cylinders. Maybe
  problems with future releases.

  Unfortunately, it looks like there won't be problems with future
  releases of the FFS and trackdisk.device.
```

E. Mounting more than one drive per unit
~~~~~~
When using the FastFileSystem it isn't possible to mount more than
one drive per unit (diskdrive). If you use other filesystems, which
don't use TD_REMOVE but TD_ADDCHANGEINT instead it's possible to
mount more than one.

F. The LoadWB Problem
~~~~~~~~~~~~~~~~~~~~~~~
If a drive is mounted after LoadWB, there will be no icon on the
workbench for unformatted or NDOS disk. This is an oddity of
the workbench and the same for ALL devices/filesystems (e.g.
CrossDOS).

G. The Disk Problem
~~~~~~~~~~~~~~~~~~~~~
Since some friends and I have had some problems with NoName
disks I have a request:

If you test DSD and a disk can't be formatted, please test
whether it can be formatted with the trackdisk.device (DFx).

Especially watch out for cheap HD disks. I bought 40 of them
and tried to format them but it wasn't possible (with several
drives). Before I tried to get them exchanged, I tried to
format them on my PC. And what a marvel, they worked fine.
How could this be ? The PC HD drives double the writing speed
and the Amiga HD drives half the motor speed. It looks like
some (very cheap) disks are so badly coated that they don't
work. See also induction.

In general, in the meantime the NoName disks are so bad that
it isn't profiteable anymore to buy that rubbish.
And I don't say this to get as less trouble with my device as
possible but since it's true. In the past, the quality of the
'NoNames' was much better. There was nearly no waste but in
the last time it was about 10-20%. I won't buy any.

H. The DirUtil Problem
~~~~~~~~~~~~~~~~~~~~~~~~
Some people told me that the 'freespace' of DSD disks isn't
displayed correctly with their dirutil. I tested some of them
(dirutils not people) and found one which displays the wrong
size. But not only with DSD disks but generally. It was a
commercial product (but NOT Directory Opus, at least not the
version I tested).
You easily can test your dirutil. Open a shell and type 'info
DSx:' (x=0-3). In the output there is a column 'Free'. Under
this the value of free blocks is displayed. Multiply this val.
by 512 (or if you're using another block size with your size)
and you get the free bytes on the disk. Your dirutil should
display the same value.

```
I. The '????' Problem
~~~~~~~~~~~~~~~~~~~~~~
Recently some people complained that there's always a 'DFx:????'
icon when inserting a DSD disk.
DSD has nothing to do with that. It's only the (normal) sign for
a disk in the drive DFx: which can't be read. When using DSD you
have two logical drives which share one physical drive. If you
use CrossDOS too, you have three log. drives which share the same
phy. drive (e.g. DF0:, DS0: and PC0:).

If you only don't like that, the ????-icon looks like the normal
disk icon, you can easily change it yourself.

Run 'IconEdit' (System:Tools/). Change the icontype to 'disk'.
Now draw an icon you want to see for a nonreadable disk (e.g.
a canceled disk, a skull, a bunch of flowers or anything).
Save it as 'Default Icon'. IconEdit saves it once as

SYS:Prefs/Env-Archive/Sys/def_disk.info

and also as

RAM:ENV/Sys/def_disk.info.

Rename both of them from def_disk.info to def_kick.info. When
you insert the next unreadable disk, you will see your creation
instead of the standard disk icon. This will definitely help you
find the readable disks.

CAUTION: If you already have a default disk icon, you have to
store it somewhere because it would be overwritten. Copy it back
after the new ones have been renamed as def_kick.info.

If you don't like that there're two icons for one (physical)
drive, you can use the MFS of Nicola Salmoria. After installing
'MultiFileSystem' you have only one drive 'DF0:' which handles
DF0:, DS0:, PC0: or everything you want.


J. The Copy program Problem
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The only problem (?) is that there's no copy program which copies
DSD disks FAST. So you have to use the normal 'DiskCopy' program
which came with your machine. If you think it's too slow (although
it isn't), you can use the 'NoVerify' option. Example: 'diskcopy
DS0: to DS1: noverify' copies the disk in unit 0 to the disk in
unit 1 WITHOUT verify.
You should use this with quality disks, only !!

The nibble mode of some programs copies DSD disks but it's surely
slower than DiskCopy WITH verify.


K. The Format Problem
~~~~~~~~~~~~~~~~~~~~~~
The 'Format' program of Workbench 2.1 and higher has besides the
bug that the display (in the shell) is completely wrong and
```

besides the bug that it flips out if TD_FORMAT returns an error
(according to the docs, TD_FORMAT returns an errorcode !) one
really nasty bug.
If you format a disk which was already formatted AND has a read/
write error AND the next cylinder is ok, IT happens. Detailed:
The format program first formats and then reads all cylinders of
a disk (as you may know). If the (verify)read returns an error, it
doesn't retry (as earlier versions do) BUT it reads (!¿?) the next
cylinder.

-> If the next cylinder is ok, no error is returned. It continues
   formatting the disk !
-> Storing data on such a disk is nothing but a game of chance.

This bug happens with ALL devices ! Also with trackdisk disks !
Fortunately (?) you get an error when copying files to the disk.
The most disgusting thing about this bug was that it cost me many
hours of work. Because I searched it in my device. It nearly (?)
drove me crazy.

Remedy: Use the format program of Workbench 2.0 (it doesn't look
  that good, but it works!). Or clear your disks with a
  copy program. Or format disks which you want to use as DSD
  disks with 'SuperDuper' without verifying and format them
  as usual.

1.12   hints

Hints
~~~~~
  This section will be updated with your hints. So if you've
  found something which will help others too, please send it
  to me (please don't phone).

  When mounting with Workbench 2.0 it's better to use the line
  'AddBuffers DSx: >NIL:' in the user-startup instead of the
  'Mount=1' line in the mountlist. If you use 'Mount=1', you
  get a 'No Disk in Drive XXX' requester.

  You want to copy some data disks to the new format: The fastest
  way to do this, is to mount a RamDisk with the same size as a
  DSD disk. Sample Mountfiles are included (DD/HD drawer). Copy
  the files to the RamDisk and after it is full, 'DiskCopy' it
  to DSx:. Example: 'diskcopy rdd: to ds0:' copies the 984K ramdisk
  to DS0: (drive0). The only thing you need for this method is
  1 (DD) or 2 (HD) megs of ram (for the ramdisk only!).

  If there are problems or you simply don't like that DSD accesses
  disks during a harddrive backup, 'diskcopy' them (or it). Example:
  you have two drives DS0: and DS1: and you're backing up on DF0:
  and DF1: then open a shell and type 'diskcopy ds0: to ds1:'.
  Diskcopy asks you to press <return> or <Ctrl-C> but simply do
  nothing. The effect is, that the drives get 'inhibited', this
  means that a diskchange isn't noticed. This 'trick' works also
  vice versa (inhibit DFx when using DSx). It also works with

```
only one drive 'diskcopy ds0: to ds0:'.
```

```
You can't boot from DSD disks directly. But there's a possibility
to do it indirectly. It can be used to boot from ANY device
with ANY filesystem. Make a normal (not DSD) bootdisk. Use this
disk to mount a RAD: with about 30K. Copy all files you need
to mount DSD to this RAD: (you needn't call it RAD:). Redirect
booting to the DSD disk. #
If you want to boot with another filesystem, you simply have to
enlarge the RAD: and copy the filesystem, too.
This is really indirect but very flexible. You can use it to
boot from ANYTHING (e.g. parnet !).
Since it isn't easy to make such a bootdisk, I've included a
script. See 'Support/MakeBootDisk'.
```

```
Here you could write your hint.
```

1.13   adresse

```
Address
~~~~~~~
  Klaus Deppisch
  Beckenwengert 3
  97246 Eibelstadt
  Germany

  Tel.: 09303 1579
```

1.14   information

```
Information
~~~~~~~~~~~

  A. Safety
  ~~~~~~~~~
DSD has been tested since Dec 92 on all Amigas available (500
- 4000).
And it is (at least with some friends and I) used. We have about
800 DSD disks and no problems. You see, it's in my own interest,
to make DSD as safe as the trackdisk.device.

If you switch on the new 'VerifyWrite' mode, DSD is safer than
the trackdisk.device. But writing speed is nearly halved (nothing
is for free).


  B. Speed
  ~~~~~~~~
DSD (V3.0) is noticeable faster than the trackdisk.device (DFx).
Tests under V37 (KickStart 2.0) FastFileSystem using DiskSpeed
V4.2.
```

```
disk/mode diskspare    trackdisk

DD    w    up to 12300 Bytes/sec up to 10900 Bytes/sec
DD    r    up to 25000 Bytes/sec up to 22700 Bytes/sec
HD    w    up to 13500 Bytes/sec up to 11600 Bytes/sec
HD    r    up to 26400 Bytes/sec up to 23500 Bytes/sec
```

The only thing where the trackdisk.device is better than DSD is
the CPU usage during data transfer. DSD needs about 10-20% more
CPU time than trackdisk does.

     C. Viruses
     ~~~~~~~~~~
     If you're using DSD disks, you can forget bootblock viruses.
     If they could write themselves on a DSD bootblock, it was their
     last activity. You can't boot from DSD disks so the virus won't
     be executed anymore.
     Also if you use the indirect boot method (via RAD:), the boot
     blocks won't be loaded. Unfortunately link viruses work on DSD
     disks, too.

1.15   pfs

```
Professional File System
~~~~~~~~~~~~~~~~~~~~~~~~~
  Version 9.5 - 10.10.94 (e.g. AMINET)
```

  This Version of PFS contains a patched version of DSD V2.2.
  Patched by Michiel Pelt. WITHOUT my knowledge ! CONTRARY to my
  (and his, too !) distribution conditions !

  Because of this NASTINESS, I'm definitely not interested in
  cooperating with Michiel Pelt anymore.

  It is expressly forbidden for Michiel Pelt to distribute my
  device, in ANY version, with his PFS from November 94 on.

  In comparison with older versions, PFS Version 9.5 doubles the
  writing speed.
  Since I don't want that this causes dataloss with my device, the
  new VerifyWrite mode MUST be switched on when using PFS V9.5.
  The included Mountfiles are already changed.

  I STRONGLY recommend you not to use older DSD versions than 3.0
  (without the VerifyWrite mode) with PFS versions higher or equal
  than 9.4.

  Version 7.5 - 20.4.94

  DSD Version 3.0 doesn't work with PFS Version 7.5 on 68000 Amigas.
  If you think you have to use PFS, try to get Version 9.5.