Restoring damaged partitions with dd_rescue

dd to the rescue

If you backup your data at regular intervals, you have nothing to worry about in case of hard disk failure. However, what happens if you forget that all-important backup? Murphy's Law states that disk read errors are likely to occur just then. A good thing if you have "dd_rescue" in your toolkit. BY HANS-GEORG EßER

There are any number of potential causes of hard disk failure. If a head crash destroys your disk's read/write heads, you will need help from an expert data recovery service.

The damage is typically less severe: a few sectors may be unreadable, and the disk's recovery mechanism, which should have a few sectors in reserve to compensate for the damage, may simply have run out of space.

If you are lucky, you may still be able to mount the partitions on a damaged hard disk, and the error messages may be restricted to just a few files (or even a single file). In this case you can simply copy the files from the damaged partition to a new hard disk, and dispose of the old disk, or ask the manufacturer to replace it under warranty, when you are finished. Whatever you do, avoid using a disk that you know is damaged. Although you can use *badblocks* (see Box 1) to explicitly hide damaged sectors, the danger of the disk dying on you is just too big.

Mount Failure

Scenarios where you are unable to mount the damaged partition are far more serious. The *mount* command will fail if the management information at the start of the position is missing, and *fsck* is unable to remedy the condition because it cannot write to the bad blocks (see Box 2).

You have one option in this scenario: you need to create a copy of the filesystem and store it on a second disk. Then go on to modify the image file using *fsck*. As filesystems have copies of the management information at various places, you may be able to recover your data.

Knoppix: dd_rescue On Board

The *dd_rescue* tool by Kurt Garloff [1] is a variant of the legacy Unix tool **dd**, which was developed with data recovery in mind. Knoppix includes *dd_rescue*, so Knoppix users can avoid building the tool from the source code. Users of other operating systems should check out the *dd_rescue* homepage for the sources. We will be discussing the Knoppix approach.

Box 1: Checking Media with badblocks

As the name would suggest, the *badblocks* tool checks for bad blocks on media such as floppies and hard disks. You can use *badblocks* in combination with *fsck* to tag the damaged sectors of the filesystem as unusable. To do so, pass the output from *badblocks* to *fsck*:

```
badblocks /dev/hdb5 > ➥
/tmp/bad-blocks
fsck -t ext2 -l ➥
/tmp/bad-blocks /dev/hdb5
```

If you are reformatting a damaged disk, there is no need to call *badblocks*. The *mkfs* commands have a *-c* (check) option that automagically runs *badblocks* before formatting:

```
mke3fs -c /dev/hdb5
```

INFO

[1] dd_rescue, *http://www.garloff.de/kurt/linux/ddrescue/*

Besides ease of use, *dd_rescue* has two big advantages in comparison to the older *dd*:

- If a read error occurs during copying, *dd_rescue* will not simply abort the copy. Instead the tool writes a string of null bytes with the same size as the bad sector to the target file. This gives you a complete partition image, apart from the null bytes in the unreadable sectors.
- *dd_rescue* can use two block sizes for read access. In error free areas of the disk, it reads larger blocks that default to 16384 bytes; in case of error, the tool reduces the read size to a default of 512 bytes. You can change both of these values.

GLOSSARY

dd: *The legacy Unix tool "dd" copies files between block devices and/or regular files. The tool can be used to write a floppy image to a disk (the developers would have preferred to call it "cc" for "copy and convert" – but this mnemonic was already in use by the C compiler).*

First Steps

To get things ready, you need to install the hard disk in your machine. You also need another disk with a free partition big enough to handle the steps that follow. The partition should have twice the capacity of the partition on the damaged disk that you will be attempting to recover .

After booting Knoppix, pop up a console window and change users to *root* by typing *sudo su* (no password required). Then run *fdisk -l* to output a list of partitions.

After identifying the damaged partition, you can start the recovery operation. First, mount a partition on the second (error-free) disk, in */mnt*, for example:

```
mount /dev/hdb1 /mnt
```

Assuming that */dev/hda5* is the damaged partition, you would need the following *dd_rescue* command:

```
dd_rescue -A /dev/hda5 /mnt/image.dat
```

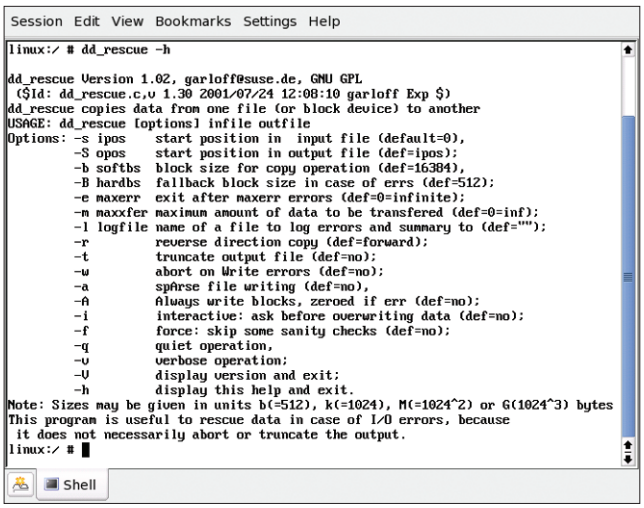


Figure 1: Running "dd_rescue" with the "-h" parameter gives you a list of options.

Over 10,000 Red Hat Certified Engineers

The leading certification for Linux has reached another milestone: over 10,000 RHCEs worldwide, and over 3,500 RHCTs. RHCE certification includes both 'hands-on'and theoretical training. Students must 'prove what they can do' in order to be certified.

redhat CERTIFIED TECHNICIAN

RED HAT CERTIFIED TECHNICIAN
Core sys admin skills

- install and configure new Red Hat systems
- attach new systems to an existing network
- perform core system administration

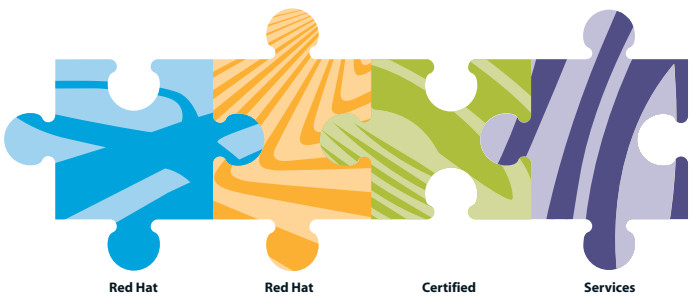
Starting from scratch? Our new RHCT certification may be just what you need. Fewer required classes. Same legendary hands-on training and performance-based testing. As a Red Hat Certified Technician, you'll learn what you need to know to get the job done. And RHCT is the first step towards RHCE.

redhat CERTIFIED ENGINEER

RED HAT CERTIFIED ENGINEER
Senior sys admin skills

- RHCT-level skills plus:
- set up networking services
- configure security
- diagnostics and troubleshooting

Want to be a Linux guru? RHCE is the gold standard certification. According to Certification Magazine/Fairfield Research. RHCE is the #1 certification for overall quality of program, #1 for quality of tests and exams, and #1 for overall educational quality.



Tel: 01483 734909
www.europe.redhat.com/training
Email: training-uk@redhat.com

redhat

© 2004 Red Hat, Inc. All rights reserved. "Red Hat," Red Hat Linux, the Red Hat "Shadow Man" logo, and the products listed are trademarks or registered trademarks of Red Hat, Inc. in the US and other countries. Linux is a registered trademark of Linus Torvalds.

Box 2: Hard Disk Errors

Read errors that prevent a partition from mounting are reflected by kernel log entries as shown next:

```
Feb  6 05:28:42 server kernel: reiserfs: found format "3.6" with standard journal
Feb  6 05:28:47 server kernel: reiserfs: enabling write barrier flush mode
Feb  6 05:28:47 server kernel: reiserfs: using ordered data mode
Feb  6 05:28:50 server kernel: hdb: dma_intr: status=0x51 { DriveReady SeekComplete Error }
Feb  6 05:28:50 server kernel: hdb: dma_intr: error=0x40 { UncorrectableError }, LBAsect=19616787, ➥
sector=65680
Feb  6 05:28:50 server kernel: end_request: I/O error, dev 03:42 (hdb), sector 65680
Feb  6 05:28:50 server kernel: journal-459: unable to read  journal header
Feb  6 05:28:50 server kernel: sh-2022: reiserfs_read_super: unable to initialize journal space
```

There is no need for the typical *dd* parameters, *if=* and *of=*, with *dd_rescue*. The first parameter specifies the source, and the second parameter the target. The program displays a progress indicator while it is running, and it outputs a warning if it detects an error condition. Box 3 shows a sample *dd_rescue* session including the command syntax and output.

In case of minimal damage to the disk, you should have an image of the damaged partition a short time later. The image will have any readable data at the right place on the partition.

Pick Up the Pieces

If a larger area of the disk is damaged, *dd_rescue* may not make any progress at a certain point as it can see only bad blocks. In this case, your only option is to quit, and re-run *dd_rescue* using a different target file, and stipulating the *-r* option:

```
dd_rescue -A -r /dev/hda5➥
 /mnt/image-tail.dat
```

This tells *dd_rescue* to start at the end of the partition and work backwards towards the damaged sectors. Of course, the tool is bound to hit the damaged sectors after a while, and you will have to quit again. You can then compare the size of the two files, *image.dat* and *image-tail.dat*, with the size of the original partition to discover the size of the missing section. Create an empty file of the right size by typing:

```
dd if=/dev/zero➥
 of=/mnt/image-zero.dat ➥
 bs=1024 count=N
```

and put all the pieces together to form an image:

```
cd /mnt; cat image.dat➥
 image-zero.dat image-tail.dat➥
 > image-full.dat
```

Repair Work

Before you start to work on the image file that *dd_rescue* has created, it makes sense to back up the image in case anything goes wrong. This is why I recommended a disk with twice the space of the original partition earlier on:

```
cp /mnt/image.dat➥
 /mnt/image.dat.copy
```

You can then use *fsck* with the appropriate options to repair the recovered image file. If you have a partition from a Reiser filesystem, the command would be as follows:

```
reiserfsck --fix-fixable➥
 /mnt/image.dat
```

or

```
fsck.ext3 -p /mnt/image.dat
```

for an Ext3 filesystem.

Then mount the repaired system just to make sure (choose the read-only option):

```
mkdir /temp
mount -o loop,ro➥
 /mnt/image.dat /temp
```

You should now be able to access the data on the damaged partition in the */temp* directory and copy any data you require to another directory: *cp -a /temp TARGET*. Alternatively, use

```
rsync -av --rsh="ssh"➥
 /temp root@computer:TARGET/
```

to send a copy across your network to another computer. ■

Box 3: dd_rescue output

```
01 root@ttyp0[knoppix]# dd_rescue -A /dev/hda5 /mnt/image.dat
02 dd_rescue: (info): ipos:    859444.5k, opos:    859444.5k, xferd:    859444.5k
03                    errs:       0, errxfer:         0.0k, succxfer:    859444.5k
04                +curr.rate:    30166kB/s, avg.rate:    41084kB/s, avg.load: -0.6%
05 dd_rescue: (warning): /dev/hda5 (859444.5k): Input/output error!
06 dd_rescue: (info): ipos:  80043264.0k, opos:  80043264.0k, xferd:  80043264.0k
07                    errs:       1, errxfer:         0.5k, succxfer:  80043263.5k
08                +curr.rate:    30166kB/s, avg.rate:    41084kB/s, avg.load: -0.6%
09 dd_rescue: (info): /dev/hda5 (80043264.0k): EOF
10 Summary for /dev/hda5 -> /mnt/image.dat:
11 dd_rescue: (info): ipos:  80043264.0k, opos:  80043264.0k, xferd:  80043264.0k
12                    errs:       0, errxfer:         0.5k, succxfer:  80043263.5k
13                +curr.rate:        0kB/s, avg.rate:    41084kB/s, avg.load: -0.6%
14 root@ttyp0[knoppix]# _
```