WMLUG May 2015

Introduction to ddrescue

by

Patrick TenHoopen

What is ddrescue?

ddrescue is a utility for copying data from one file or block device to another and is part of the GNU project. It was originally written by Antonio Diaz Diaz and first released in 2004.

ddrescue is available in repositories, at www.gnu.org/s/ddrescue/, and on bootable rescue CDs such as SystemRescueCD.

Why/when use it?

You would use it when errors are detected on a hard drive from bad sectors. By running it you are attempting to create a copy of the affected data to be able to write it to a new disk or partition to run subsequent file system checks on it.

Whereas the venerable dd will exit when an error is encountered, ddrescue attempts to read the data, multiple times if configured to do so, so that as much data can be recovered as possible.

Why/when use it? (cont.)

It can be run multiple times to try to fill in missing data since it reuses the same output image file.

If you have multiple damaged copies of the data, ddrescue can read from them all to try to fill in the missing data, using the same output file.

Why/when to not use it?

If the disk is having mechanical issues (emitting clicking noises, grinding, etc.), accessing the data could destroy more of it.

ddrescue v dd_rescue

ddrescue and dd_rescue are different programs that don't share any code.

dd_rescue, written by Kurt Garloff, was developed for the same reason as ddrescue. It also doesn't quit if an error is encountered.

ddrescue v dd_rescue (cont.)

ddrescue supersedes some of dd_rescue's functionality but dd_rescue is "still useful for some special features, such as direct IO, sparse copies, splice copies, and preallocation" (according to its website).

In Debian and Ubuntu, you will want the gddrescue package, not ddrescue which is actually dd_rescue.

Usage Notes

You can run it against an entire disk or just the partitions you need. The destination must have enough free space to hold the image. The image will be the size of the device and not just the used space.

If a device or partition is specified as the destination, all data on it will be overwritten.

Usage Notes (cont.)

It could take a long time (possibly days) for ddrescue to recover the data depending on how many bad sectors it finds and the number of retries it is set to do.

If the drive gets too hot, it may become less reliable. If this happens, stop ddrescue, let the drive cool off and restart ddrescue which will continue on the drive where it left off.

Usage Syntax

```
ddrescue [options] infile outfile [logfile]
```

You should use a logfile unless you know what you are doing.

If you reboot, check the device names before restarting ddrescue.

Do not use options '-F' or '-g' without reading the manual first.

Options

```
-h --help                      display this help and exit
-V, --version                  output version information and exit
-a, --min-read-rate=<bytes>    minimum read rate of good areas in bytes/s
-A, --try-again                mark non-split, non-trimmed blocks as non-
tried
-b, --sector-size=<bytes>      sector size of input device [default 512]
-B, --binary-prefixes          show binary multipliers in numbers [SI]
-c, --cluster-size=<sectors>   sectors to copy at a time [128]
-C, --complete-only            do not read new data beyond logfile limits
-d, --direct                   use direct disc access for input file
-D, --synchronous              use synchronous writes for output file
```

Options (cont.)

```
-e, --max-errors=[+]<n>        maximum number of [new] error areas allowed
-E, --max-error-rate=<bytes>   maximum allowed rate of read errors per
second
-f, --force                    overwrite output device or partition
-F, --fill-mode=<types>        fill given type blocks with infile data (?
*/-+)
-g, --generate-mode            generate approximate logfile from partial
copy
-i, --input-position=<bytes>   starting position in input file [0]
-I, --verify-input-size        verify input file size with size in logfile
-K, --skip-size=<bytes>        initial size to skip on read error [64 KiB]
-l, --logfile-size=<entries>   do not grow logfile beyond this size [1000]
-m, --domain-logfile=<file>    restrict domain to finished blocks in file
```

Options (cont.)

```
-M, --retrim                    mark all failed blocks as non-trimmed
-n, --no-split                  do not try to split or retry failed
blocks
-o, --output-position=<bytes>   starting position in output file [ipos]
-p, --preallocate               preallocate space on disc for output file
-q, --quiet                     suppress all messages
-r, --retries=<n>               exit after given retries (-1=infinity)
[0]
-R, --reverse                   reverse direction of copy operations
-s, --size=<bytes>              maximum size of input data to be copied
-S, --sparse                    use sparse writes for output file
-t, --truncate                  truncate output file to zero size
```

Options (cont.)

```
-T, --timeout=<interval>       maximum time since last successful read
-v, --verbose                  be verbose (a 2nd -v gives more)
-w, --ignore-write-errors      make fill mode ignore write errors
-x, --extend-outfile=<bytes>   extend outfile size to be at least this
long
```

Example Run Part 1

First copy as much data as possible, without retrying or splitting sectors:

```
ddrescue --no-split /dev/hda1
/media/sdc1/hda_image_file.img
/media/sdc1/hda_log_file.log
```

ddrescue Example Output

```
GNU ddrescue 1.17

Press Ctrl-C to interrupt

Current status

rescued:    78529 MB,  errsize:       0 B,  current rate:   19595 kB/s

   ipos:    78529 MB,   errors:       0,    average rate:   27820 kB/s

   opos:    78529 MB,    time since last successful read:       0 s

Copying non-tried blocks...
```

Example Run Part 2

If it found errors, now let it retry reading those locations that it previously found (and recorded in the logfile) 3 times, using uncached reads:

```
ddrescue --direct --max-retries=3 /dev/hda1
/media/sdc1/hda_image_file.img
/media/sdc1/hda_log_file.log
```

Example Run Part 3

If that fails you can try again but using the retrimmed option, so it tries to reread full sectors:

```
ddrescue --direct --retrim --max-retries=3
/dev/hda1 /media/sdc1/hda_image_file.img
/media/sdc1/hda_log_file.log
```

Mounting the Recovered Image

To mount an image containing one partition:

```
mount -o loop,ro imagefile mountpoint
```

To mount an image containing multiple partitions, specify the offset of the partition you want:

```
mount -o loop,ro,offset=9999 imagefile mountpoint
```

Fixing the Image

Once the image is created, utilities can be used to fix issues:

- fsck
- testdisk

Recovering Data

Data can be recovered from the image file with utilities:

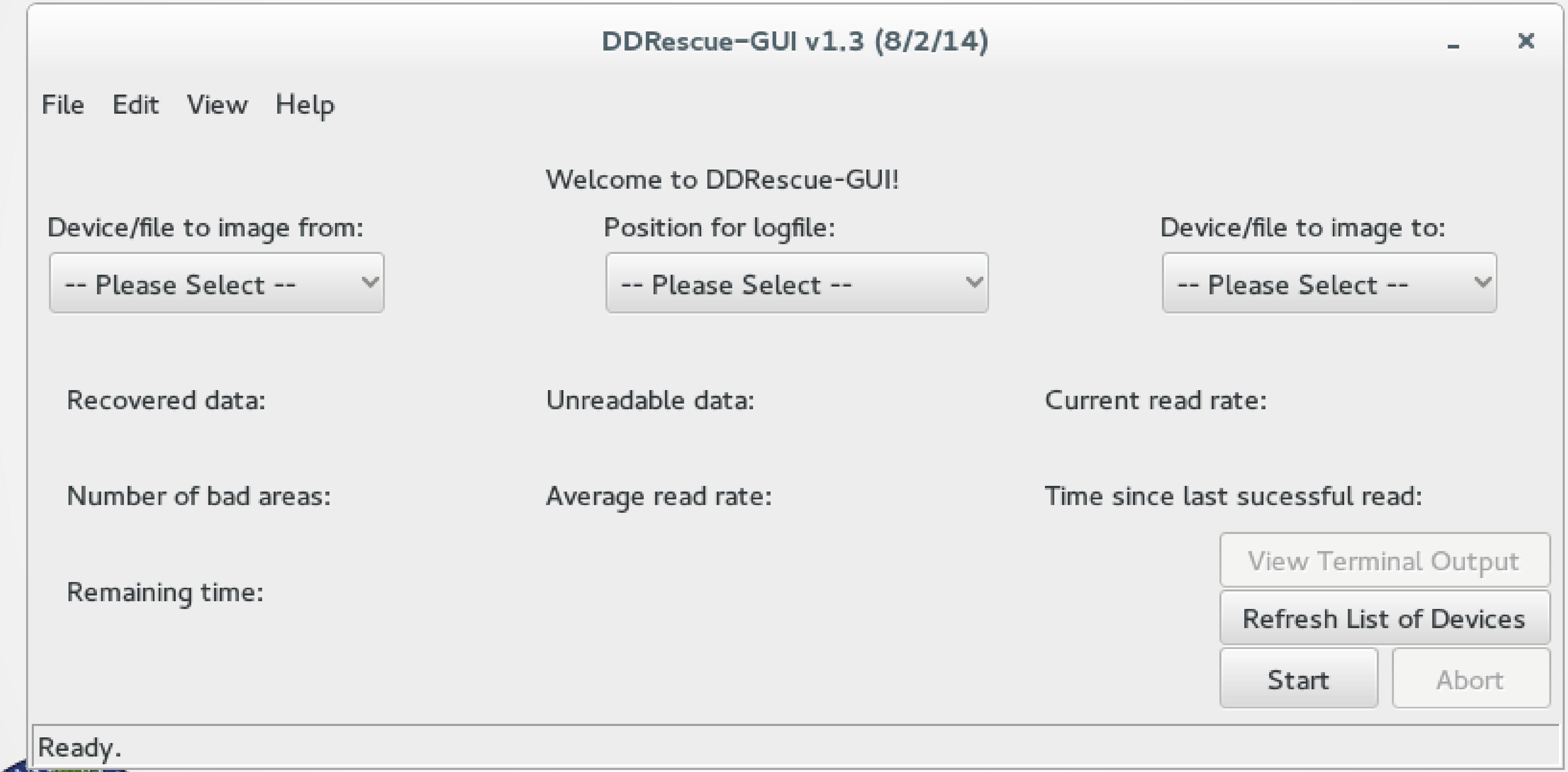
- photorec

DDRescue-GUI

DDRescue-GUI is a program written by Hamish McIntyre-Bhatty designed to make it easier to use GNU ddrescue, providing a simple graphical interface written in Python 2.

It is available at: https://launchpad.net/ddrescue-gui

DDRescue-GUI



DDRescue-GUI v1.3 (8/2/14)

Welcome to DDRescue-GUI!

Device/file to image from: -- Please Select --

Position for logfile: -- Please Select --

Device/file to image to: -- Please Select --

Recovered data:

Unreadable data:

Current read rate:

Number of bad areas:

Average read rate:

Time since last sucessful read:

Remaining time:

Ready.

Guides

Guide to Using DDRescue to Recover Data

https://www.technibble.com/guide-using-ddrescue-recover-data/