Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'e2image.8'

$ man e2image.8

E2IMAGE(8)   System Manager's Manual   E2IMAGE(8)

NAME

 e2image - Save critical ext2/ext3/ext4 file system metadata to a file

SYNOPSIS

 e2image [-r|-Q [-af]] [-b superblock] [-B blocksize] [-cnps] [-o

 src_offset] [-O dest_offset] device image-file

 e2image -I device image-file

DESCRIPTION

 The e2image program will save critical ext2, ext3, or ext4 file system

 metadata located on device to a file specified by image-file. The im?

 age file may be examined by dumpe2fs and debugfs, by using the -i op?

 tion to those programs. This can assist an expert in recovering cata?

 strophically corrupted file systems.

 It is a very good idea to create image files for all file systems on a

 system and save the partition layout (which can be generated using the

 fdisk -l command) at regular intervals --- at boot time, and/or every

 week or so. The image file should be stored on some file system other

 than the file system whose data it contains, to ensure that this data

is accessible in the case where the file system has been badly damaged.

To save disk space, e2image creates the image file as a sparse file, or in QCOW2 format. Hence, if the sparse image file needs to be copied to another location, it should either be compressed first or copied using the --sparse=always option to the GNU version of cp(1). This does not apply to the QCOW2 image, which is not sparse.

The size of an ext2 image file depends primarily on the size of the file systems and how many inodes are in use. For a typical 10 Gigabyte file system, with 200,000 inodes in use out of 1.2 million inodes, the image file will be approximately 35 Megabytes; a 4 Gigabyte file system with 15,000 inodes in use out of 550,000 inodes will result in a 3 Megabyte image file. Image files tend to be quite compressible; an im‐ age file taking up 32 Megabytes of space on disk will generally com‐ press down to 3 or 4 Megabytes.

If image-file is -, then the output of e2image will be sent to standard output, so that the output can be piped to another program, such as gzip(1). (Note that this is currently only supported when creating a raw image file using the -r option, since the process of creating a normal image file, or QCOW2 image currently requires random access to the file, which cannot be done using a pipe.

OPTIONS

-a   Include file data in the image file. Normally e2image only in‐ cludes fs metadata, not regular file data.   This option will produce an image that is suitable to use to clone the entire FS or for backup purposes. Note that this option only works with the raw (-r) or QCOW2 (-Q) formats. In conjunction with the -r option it is possible to clone all and only the used blocks of one file system to another device/image file.

-b superblock

Get image from partition with broken primary superblock by using the superblock located at file system block number superblock. The partition is copied as-is including broken primary su‐ perblock.

-B blocksize

> Set the file system blocksize in bytes. Normally, e2image will
> search for the superblock at various different block sizes in an
> attempt to find the appropriate blocksize. This search can be
> fooled in some cases. This option forces e2fsck to only try lo?
> cating the superblock with a particular blocksize. If the su?
> perblock is not found, e2image will terminate with a fatal er?
> ror.

-c   Compare each block to be copied from the source device to the
> corresponding block in the target image-file. If both are al?
> ready the same, the write will be skipped. This is useful if
> the file system is being cloned to a flash-based storage device
> (where reads are very fast and where it is desirable to avoid
> unnecessary writes to reduce write wear on the device).

-f   Override the read-only requirement for the source file system
> when saving the image file using the -r and -Q options.   Nor?
> mally, if the source file system is in use, the resulting image
> file is very likely not going to be useful. In some cases where
> the source file system is in constant use this may be better
> than no image at all.

-I   install the metadata stored in the image file back to the de?
> vice.   It can be used to restore the file system metadata back
> to the device in emergency situations.

WARNING!!!! The -I option should only be used as a desperation measure
when other alternatives have failed. If the file system has changed
since the image file was created, data will be lost. In general, you
should make another full image backup of the file system first, in case
you wish to try other recovery strategies afterward.

-n   Cause all image writes to be skipped, and instead only print the
> block numbers that would have been written.

-o src_offset

> Specify offset of the image to be read from the start of the
> source device in bytes. See OFFSETS for more details.

-O tgt_offset

    Specify offset of the image to be written from the start of the
    target image-file in bytes. See OFFSETS for more details.

-p   Show progress of image-file creation.

-Q   Create a QCOW2-format image file instead of a normal image file,
    suitable for use by virtual machine images, and other tools that
    can use the .qcow image format. See QCOW2 IMAGE FILES below for
    details.

-r   Create a raw image file instead of a normal image file. See RAW
    IMAGE FILES below for details.

-s   Scramble directory entries and zero out unused portions of the
    directory blocks in the written image file to avoid revealing
    information about the contents of the file system. However,
    this will prevent analysis of problems related to hash-tree in‐
    dexed directories.

RAW IMAGE FILES

    The -r option will create a raw image file, which differs from a normal
    image file in two ways. First, the file system metadata is placed in
    the same relative offset within image-file as it is in the device so
    that debugfs(8), dumpe2fs(8), e2fsck(8), losetup(8), etc. and can be
    run directly on the raw image file. In order to minimize the amount of
    disk space consumed by the raw image file, it is created as a sparse
    file.   (Beware of copying or compressing/decompressing this file with
    utilities that don't understand how to create sparse files; the file
    will become as large as the file system itself!) Secondly, the raw im‐
    age file also includes indirect blocks and directory blocks, which the
    standard image file does not have.

    Raw image files are sometimes used when sending file systems to the
    maintainer as part of bug reports to e2fsprogs. When used in this ca‐
    pacity, the recommended command is as follows (replace hda1 with the
    appropriate device for your system):

        e2image -r /dev/hda1 - | bzip2 > hda1.e2i.bz2

    This will only send the metadata information, without any data blocks.

However, the filenames in the directory blocks can still reveal infor?

mation about the contents of the file system that the bug reporter may

wish to keep confidential. To address this concern, the -s option can

be specified to scramble the filenames in the image.

Note that this will work even if you substitute /dev/hda1 for another

raw disk image, or QCOW2 image previously created by e2image.

QCOW2 IMAGE FILES

The -Q option will create a QCOW2 image file instead of a normal, or

raw image file. A QCOW2 image contains all the information the raw im?

age does, however unlike the raw image it is not sparse. The QCOW2 im?

age minimize the amount of space used by the image by storing it in

special format which packs data closely together, hence avoiding holes

while still minimizing size.

In order to send file system to the maintainer as a part of bug report

to e2fsprogs, use following commands (replace hda1 with the appropriate

device for your system):

    e2image -Q /dev/hda1 hda1.qcow2

    bzip2 -z hda1.qcow2

This will only send the metadata information, without any data blocks.

As described for RAW IMAGE FILES the -s option can be specified to

scramble the file system names in the image.

Note that the QCOW2 image created by e2image is a regular QCOW2 image

and can be processed by tools aware of QCOW2 format such as for example

qemu-img.

You can convert a .qcow2 image into a raw image with:

    e2image -r hda1.qcow2 hda1.raw

This can be useful to write a QCOW2 image containing all data to a

sparse image file where it can be loop mounted, or to a disk partition.

Note that this may not work with QCOW2 images not generated by e2image.

OFFSETS

Normally a file system starts at the beginning of a partition, and

e2image is run on the partition. When working with image files, you

don't have the option of using the partition device, so you can specify

the offset where the file system starts directly with the -o option. Similarly the -O option specifies the offset that should be seeked to in the destination before writing the file system.

For example, if you have a dd image of a whole hard drive that contains an ext2 fs in a partition starting at 1 MiB, you can clone that image to a block device with:

    e2image -aro 1048576 img /dev/sda1

Or you can clone a file system from a block device into an image file, leaving room in the first MiB for a partition table with:

    e2image -arO 1048576 /dev/sda1 img

If you specify at least one offset, and only one file, an in-place move will be performed, allowing you to safely move the file system from one offset to another.

AUTHOR

e2image was written by Theodore Ts'o (tytso@mit.edu).

AVAILABILITY

e2image is part of the e2fsprogs package and is available from http://e2fsprogs.sourceforge.net.

SEE ALSO

dumpe2fs(8), debugfs(8) e2fsck(8)

E2fsprogs version 1.46.5   December 2021   E2IMAGE(8)