**NAME**

e2image − Save critical ext2/ext3/ext4 filesystem metadata to a file

**SYNOPSIS**

**e2image** [ **−r|Q** ] [ **−f −b** *superblock* ] [ **−B** *blocksize* ] [ **−fr** ] *device image-file*
**e2image −I** *device image-file*
**e2image −ra** [ **−cfnp** ] [ **−o** *src_offset* ] [ **−O** *dest_offset* ] *src_fs* [ *dest_fs* ]

**DESCRIPTION**

The **e2image** program will save critical ext2, ext3, or ext4 filesystem metadata located on *device* to a file specified by *image-file*. The image file may be examined by **dumpe2fs** and **debugfs**, by using the **−i** option to those programs. This can assist an expert in recovering catastrophically corrupted filesystems. In the future, e2fsck will be enhanced to be able to use the image file to help recover a badly damaged filesystem.

When saving an e2image for debugging purposes, using either the **−r** or **−Q** options, the filesystem must be unmounted or be mounted read/only, in order for the image file to be in a consistent state. This requirement can be overridden using the **−f** option, but the resulting image file is very likely not going to be useful.

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. This restriction will hopefully be lifted in a future version of **e2image**.)

It is a very good idea to create image files for all of filesystems 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 filesystem other than the filesystem whose data it contains, to ensure that this data is accessible in the case where the filesystem 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**. 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 filesystems and how many inodes are in use. For a typical 10 gigabyte filesystem, with 200,000 inodes in use out of 1.2 million inodes, the image file will be approximately 35 megabytes; a 4 gigabyte filesystem 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 image file taking up 32 megabytes of space on disk will generally compress down to 3 or 4 megabytes.

**RESTORING FILESYSTEM METADATA USING AN IMAGE FILE**

The **−I** option will cause e2image to install the metadata stored in the image file back to the device. It can be used to restore the filesystem 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 filesystem has changed since the image file was created, data **will** be lost. In general, you should make a full image backup of the filesystem first, in case you wish to try other recovery strategies afterwards.

**RAW IMAGE FILES**

The **−r** option will create a raw image file instead of a normal image file. A raw image file differs from a normal image file in two ways. First, the filesystem metadata is placed in the proper position so that e2fsck, dumpe2fs, debugfs, etc. can be run directly on the raw image file. In order to minimize the amount of disk space consumed by a raw image file, the file 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 filesystem itself!) Secondly, the raw image file also includes indirect blocks and directory blocks, which the standard image file does not have, although this may change in the future.

Raw image files are sometimes used when sending filesystems to the maintainer as part of bug reports to e2fsprogs. When used in this capacity, the recommended command is as follows (replace hda1 with the appropriate device):

        **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 information about the contents of the filesystem that the bug reporter may wish to keep confidential. To address this concern, the **−s** option can be specified. This will cause **e2image** to scramble directory entries and zero out any unused portions of the directory blocks before writing the image file. However, the **−s** option will prevent analysis of problems related to hash-tree indexed directories.

Option **−b** *superblock* can be used to get image from partition with broken primary superblock. The partition is copied as-is including broken primary superblock.

Option **−B** *blocksize* can be used to set superblock block size. Normally, e2fsck 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 locating the superblock at a particular blocksize. If the superblock is not found, e2fsck will terminate with a fatal error.

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 image does, however unlike the raw image it is not sparse. The QCOW2 image minimize the amount of disk space by storing data in special format with pack data closely together, hence avoiding holes while still minimizing size.

In order to send filesystem to the maintainer as a part of bug report to e2fsprogs, use following commands (replace hda1 with the appropriate device):

        **e2image −Q /dev/hda1 hda1.qcow2**
        **bzip2 -z hda1.qcow2**

This will only send the metadata information, without any data blocks. However, the filenames in the directory blocks can still reveal information about the contents of the filesystem that the bug reporter may wish to keep confidential. To address this concern, the **−s** option can be specified. This will cause **e2image** to scramble directory entries and zero out any unused portions of the directory blocks before writing the image file. However, the **−s** option will prevent analysis of problems related to hash-tree indexed directories.

Note that QCOW2 image created by **e2image** is 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.

Options **−b** *superblock* and **−B** *blocksize* can be used same way as for raw images.

## INCLUDING DATA

Normally **e2image** only includes fs metadata, not regular file data. The **−a** option can be specified to include all data. This will give 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 or QCOW2 formats. The **−p** switch may be given to show progress. 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), the **−c** option which cause e2image to try reading a block from the destination to see if it is identical to the block which **e2image** is about to copy. If the block is already the same, the write can be skipped. The **−n** option will cause all of the writes to be no-ops, and print the blocks that would have been written.

**OFFSETS**

> Normally a filesystem 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 filesystem starts directly with the **−o** option. Similarly the **−O** option specifies the offset that should be seeked to in the destination before writing the filesystem.

> 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 fs with:

> > **e2image −aro 1048576 img /dev/sda1**

> Or you can clone a fs 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 filesystem 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)