

LINUX ext3 FAQ

Q: Where can I find this FAQ?

Currently at: <http://batleth.sapientisat.org/projects/FAQs/ext3-faq.html>. If you have any comments or suggestions just let me know: Juri Haberland <juri-at-koschikode.com>

Do not mail me technical questions regarding ext3, please use the ext3-users mailing list (see below) for this!

Q: What is ext3?

Ext3 is a journaling filesystem developed by Stephen Tweedie. It is compatible to ext2 filesystems; actually you can look at it as an ext2 filesystem with a journal file. The journaling capability means no more waiting for fsck's or worrying about metadata corruption. What is most noticeable is that you can switch back and forth between ext2 and ext3 on a partition without any problem: it is just a matter of giving the mount command the right filesystem type.

Q: Is there a mailing list or mailing list archive?

The ext3-user mailing list is at <https://listman.redhat.com/mailman/listinfo/ext3-users/>. An archive can be found at <https://listman.redhat.com/pipermail/ext3-users/> and a searchable one at <http://marc.theaimsgroup.com/?l=ext3-users>.

Q: Where can I get ext3 for linux?

From kernel version 2.4.15 on (don't use this one!) ext3 is included in the standard kernel so you don't need any extra patch. Patches against older 2.4 kernels as well as intermediate patches can be found at <http://www.zip.com.au/~akpm/linux/ext3/>. You can get the old ext3 patches for kernel 2.2 either from your favourite kernel mirror ftp://ftp.*.kernel.org/pub/linux/kernel/people/sct/ext3/ or from <ftp://ftp.uk.linux.org/pub/linux/sct/fs/jfs/>. As of writing this FAQ the most recent version for kernel 2.2 is 0.0.7a which is rather outdated. There is no development going into ext3 for 2.2 anymore.

Q: What do I need to build an ext3 ready kernel?

From 2.4.16 on you just need a fresh kernel tarball. The following description is for kernel version 2.2, but should work in the same way for older 2.4 kernels that don't have ext3 support. You need a fresh kernel tarball and the ext3 patches:

```
http://batleth.sapientisat.org/projects/FAQs/ext3-faq.html (1 of 8) [8/4/2008 6:19:30 PM]
cd /usr/src/wget ftp://ftp.*.kernel.org/pub/linux/kernel/v2.2/linux-2.2.20.tar.bz2 wget
ftp://ftp.*.kernel.org/pub/linux/kernel/people/sct/ext3/ext3-0.0.7a.tar.bz2 rm -rf linux tar -xjvf linux-
2.2.20.tar.bz2tar -xjvf ext3-0.0.7a.tar.bz2cd linux cat ../ext3-0.0.7a/linux-2.2.19.kdb.diff | patch -sp1cat
../ext3-0.0.7a/linux-2.2.19.ext3.diff | patch -sp1
```

You will get a failed hunk in the main Makefile because of a mismatch in the kernel sublevel version. You can ignore it safely.

After this you must configure your kernel with `make [menu]config` and set `CONFIG_EXPERIMENTAL=y`. Then you can set `CONFIG_EXT3_FS=y` and build a new kernel.

You also need a recent (at least version 1.25) e2fsprogs package with ext3 support, which you can download from <ftp://download.sourceforge.net/pub/sourceforge/e2fsprogs/>.

LINUX ext3 FAQ

Q: How do I convert my ext2 partition to ext3? (was: How do I use ext3?)

Before you can mount a partition as ext3 you have to create a journal on it. The easiest way to do it is to type:

```
tune2fs -j /dev/hdaX
```

This can be done on an unmounted or on a mounted filesystem. If you create the journal on a mounted filesystem you will see a .journal file. Don't try to delete this and don't back this up or restore it from backup! If you run tune2fs -j on an unmounted partition an invisible journal file will be created.

Now you can mount the filesystem as ext3 using:

```
mount -t ext3 /dev/hdaX /mnt/somewhere
```

With mke2fs -j /dev/hdaX you can format a partition as ext3 (as always it will be also usable as ext2 partition).

Q: How can I recover (undelete) deleted files from my ext3 partition?

Actually, you can't! This is what one of the developers, Andreas Dilger, said about it:

In order to ensure that ext3 can safely resume an

<http://batleth.sapiienti-sat.org/projects/FAQs/ext3-faq.html> (2 of 8) [8/4/2008 6:19:30 PM] unlink after a crash, it actually zeros out the block pointers in the inode, whereas ext2 just marks these blocks as unused in the block bitmaps and marks the inode as "deleted" and leaves the block pointers alone. Your only hope is to "grep" for parts of your files that have been deleted and hope for the best.

Q: 'df' command says partition is full, while 'du' reports free space

Theodore Ts'o, the ext2 developer, said:

The standard cause for this is some user process keeping a deleted file open. When this happens, the space is not visible via 'du', since the file is no longer visible in the directory tree. However, the space is still used by the file until it is deallocated, and that can only happen once the last process which has the file open either closes its file descriptor to the file, or the process exits. You can use the lsof program to try to find which process is keeping an open file. Usually it's some log file, or some large data base file which gets rotated out, but some older process are still keeping the log file open.

Q: I compiled ext3 as a module and my / partition gets mounted as

<http://batleth.sapiienti-sat.org/projects/FAQs/ext3-faq.html> (3 of 8) [8/4/2008 6:19:30 PM] **ext2**
(This part was initially suggested by Boris Wesslowski; major overhaul by Theodore Ts'o)

If you have ext2 compiled into the kernel and ext3 as a module and your root filesystem is ext2/3, then the kernel will always mount the root fs as ext2 and not as ext3 since at the time when the root filesystem is mounted, the kernel does not have access to the modules, since they are located on the root filesystem. (This is a chicken and egg problem!) If you have this setup, you might first want to consider whether it makes sense to compile ext3 as a module. If you want to use ext3 on your root filesystem, the ext3 filesystem module will always be loaded and can not be unloaded, so it might as well be compiled-in. Furthermore, modules waste a tiny amount of memory (on average 2k per module), and take up an extra entry in the TLB cache --- a slight, and perhaps not measurable

LINUX ext3 FAQ

disadvantage, but given that there is no real advantage to compiling ext3 as a module, why bother? If you do want to compile ext3 as a module and use it as your root filesystem, it can be done, but what you must do is do is boot into an initial ramdisk (initrd) image as your root image. This initrd image will contain the necessary modules (scsi, ext3, etc.) so you can mount your "real" root filesystem and then use pivot_root to replace the initrd root filesystem with the "real" root filesystem. Most distributions do the pivot_root trick automatically, but they differ in how the tool which builds the initrd image needs to be called.

- On a SuSE system you have to put "jbd ext3" (in this order!) into the YaST setting INITRD_MODULES. Then do a mk_initrd and it should work.
- On a Red Hat system it seems that it is sufficient to just do a mkinitrd.
- On a Debian system (woody and above) you must run the command:

```
mkinitrd -o boot/initrd.img-2.4.18-386 /lib/modules/2.4.18-386
```

where 2.4.18-386 must be substituted with the version number of the debian kernel which you are currently using. Note: this requires that you have the initrd-tools package installed, and if you currently are not using an initrd setup, your lilo or grub configuration files must be modified to tell the booting kernel to use the initrd image.

Q: How do I convert my ext3 partition back to ext2?

Actually there is only little need to do so, because in most cases it is sufficient to mount the partition explicitly as ext2. But if you really need to convert your partition back to ext2 just do the following on an unmounted partition:

```
tune2fs -O ^has_journal /dev/hdaX
```

To be on the safe side you should force a fsck run on this partition afterwards:

```
fsck.ext2 -f /dev/hdaX
```

After this procedure you can safely delete the .journal file if there was any.

Q: If a system shutdown hard, even with journaling is it at all

<http://batleth.sapient-sat.org/projects/FAQs/ext3-faq.html> (4 of 8) [8/4/2008 6:19:30 PM]

necessary to run e2fsck?

Theodore Ts'o said:

It's best to just always run e2fsck. [...] E2fsck will run the journal automatically, and if the filesystem is otherwise clean, it skip doing a full filesystem check. If the filesystem is not clean (because during the previous run the kernel noticed some filesystem inconsistencies), e2fsck will automatically do a full check if it is necessary. If you have multiple disks, fsck will run multiple e2fsck processes in parallel, thus speeding up your boot sequence than if you let the kernel replay the journal for each filesystem when it tries to mount it, since then the journal replays will be done sequentially, instead of in parallel.

Q: What is the largest possible size of an ext3 filesystem and of files on ext3?

inspired by Andreas Dilger, suggested by Christian Kujau:

LINUX ext3 FAQ

Ext3 can support files up to 1TB. With a 2.4 kernel the filesystem size is limited by the maximal block device size, which is 2TB. In 2.6 the maximum (32-bit CPU) limit is of block devices is 16TB, but ext3 supports only up to 4TB.

Q: How do I convert the journal file from version 1 to version 2?

Just type:

```
mount /dev/hdaX /mnt -o journal=update
```

<http://batleth.sapientisat.org/projects/FAQs/ext3-faq.html> (5 of 8) [8/4/2008 6:19:30 PM]
to convert your old (ext3 v0.0.3* and earlier) filesystem to the new journal format.

Q: How do I convert my root filesystem from version 1 to version 2?

Just issue a "lilo -R linux rw rootflags=journal=update" and reboot.

Q: I updated ext3 today. Got all of my mounts converted. Now on boot, I see: "EXT3-fs: mounted filesystem with ordered data mode". Is this normal?

Nigel Metheringham answered this on the ext3-users mailing list as follows:

That's fine. The EXT3-fs message is just telling you it mounted the fs OK. It's also telling you what form of journaling you are using.

ext3 has 2 formats of journal:

- version 1 - default and only possibility for ext3 releases 0.0.3* and earlier
- version 2 - default for filesystems created with 0.0.4 and later

Version 2 journals support additional semantics required to allow metadata journaling, and provide 2 new forms of journaling -ordered and writeback. Both V1 & V2 journals support data journaling (where everything that would go to disk is journaled).

The CHANGES file in the distribution says this:

New mount options:

<http://batleth.sapientisat.org/projects/FAQs/ext3-faq.html> (6 of 8) [8/4/2008 6:19:30 PM]

"mount -o journal=update" Mounts a filesystem with a Version 1 journal, upgrading the journal dynamically to Version 2.

"mount -o data=journal" Journals all data and metadata, so data is written twice. This is the mode which all prior versions of ext3 used.

"mount -o data=ordered" Only journals metadata changes, but data updates are flushed to disk before any transactions commit. Data writes are not atomic but this mode still guarantees that after a crash, files will never contain stale data blocks from old files.

LINUX ext3 FAQ

"mount -o data=writeback" Only journals metadata changes, and data updates are entirely left to the normal "sync" process. After a crash, files will may contain stale data blocks from old files: this mode is exactly equivalent to running ext2 with a very fast fsck on reboot.

Ordered and Writeback data modes require a Version 2 journal: if you do not update the journal format then only the Journalled data will be allowed.

The default data mode is Journalled for a V1 journal, and Ordered for V2.

For more information also see RedHat's release notes from their 7.2 release.

Q: How stable is ext3?

Q: Do quotas work on ext3?

Q: Does LILO work with ext3?

In my experience it is as stable as ext2. Should do so with 2.4ish kernels. I had problems with 2.2 but others didn't...

Yes.