Ext3 Filesystem

Ext3 was originally released in September 1999. Written by Stephen Tweedie for the 2.2 branch, and ported to 2.4 kernels by Peter Braam, Andreas Dilger, Andrew Morton, Alexander Viro, Ted Ts'o and Stephen Tweedie.

Ext3 is the ext2 filesystem enhanced with journalling capabilities.

Options

When mounting an ext3 filesystem, the following option are accepted:
(*) == default

ro		Mount filesystem read only. Note that ext3 will replay the journal (and thus write to the partition) even when mounted "read only". Mount options "ro,noload" can be used to prevent writes to the filesystem.
journal=update		Update the ext3 file system's journal to the current format.
journal=inum		When a journal already exists, this option is ignored. Otherwise, it specifies the number of the inode which will represent the ext3 file system's journal file.
journal_dev=devnum		When the external journal device's major/minor numbers have changed, this option allows the user to specify the new journal location. The journal device is identified through its new major/minor numbers encoded in devnum.
norecovery noload		Don't load the journal on mounting. Note that this forces mount of inconsistent filesystem, which can lead to various problems.
data=journal		All data are committed into the journal prior to being written into the main file system.
data=ordered	(*)	All data are forced directly out to the main file system prior to its metadata being committed to the journal.
data=writeback		Data ordering is not preserved, data may be written into the main file system after its metadata has been committed to the journal.
commit=nrsec	(*)	Ext3 can be told to sync all its data and metadata every 'nrsec' seconds. The default value is 5 seconds. This means that if you lose your power, you will lose as much as the latest 5 seconds of work (your filesystem will not be damaged though, thanks to the journaling). This default value (or any low value)

will hurt performance, but it's good for data-safety. Setting it to 0 will have the same effect as leaving it at the default (5 seconds).
Setting it to very large values will improve performance.

barrier=<0(*)|1>
barrier
nobarrier (*)
This enables/disables the use of write barriers in the jbd code. barrier=0 disables, barrier=1 enables. This also requires an IO stack which can support barriers, and if jbd gets an error on a barrier write, it will disable again with a warning. Write barriers enforce proper on-disk ordering of journal commits, making volatile disk write caches safe to use, at some performance penalty. If your disks are battery-backed in one way or another, disabling barriers may safely improve performance. The mount options "barrier" and "nobarrier" can also be used to enable or disable barriers, for consistency with other ext3 mount options.

orlov (*)
This enables the new Orlov block allocator. It is enabled by default.

oldalloc
This disables the Orlov block allocator and enables the old block allocator. Orlov should have better performance - we'd like to get some feedback if it's the contrary for you.

user_xattr
Enables Extended User Attributes. Additionally, you need to have extended attribute support enabled in the kernel configuration (CONFIG_EXT3_FS_XATTR). See the attr(5) manual page and http://acl.bestbits.at/ to learn more about extended attributes.

nouser_xattr
Disables Extended User Attributes.

acl
Enables POSIX Access Control Lists support. Additionally, you need to have ACL support enabled in the kernel configuration (CONFIG_EXT3_FS_POSIX_ACL). See the acl(5) manual page and http://acl.bestbits.at/ for more information.

noacl
This option disables POSIX Access Control List support.

reservation

noreservation

bsddf (*)
Make 'df' act like BSD.
minixdf
Make 'df' act like Minix.

check=none
nocheck
Don't do extra checking of bitmaps on mount.

debug
Extra debugging information is sent to syslog.

errors=remount-ro		Remount the filesystem read-only on an error.
errors=continue		Keep going on a filesystem error.
errors=panic		Panic and halt the machine if an error occurs.
		(These mount options override the errors behavior specified in the superblock, which can be configured using tune2fs.)
data_err=ignore(*)		Just print an error message if an error occurs in a file data buffer in ordered mode.
data_err=abort		Abort the journal if an error occurs in a file data buffer in ordered mode.
grpid		Give objects the same group ID as their creator.
bsdgroups		
nogrpid	(*)	New objects have the group ID of their creator.
sysvgroups		
resgid=n		The group ID which may use the reserved blocks.
resuid=n		The user ID which may use the reserved blocks.
sb=n		Use alternate superblock at this location.
quota		These options are ignored by the filesystem. They
noquota		are used only by quota tools to recognize volumes
grpquota		where quota should be turned on. See documentation
usrquota		in the quota-tools package for more details
		(http://sourceforge.net/projects/linuxquota).
jqfmt=<quota type>		These options tell filesystem details about quota
usrjquota=<file>		so that quota information can be properly updated
grpjquota=<file>		during journal replay. They replace the above
		quota options. See documentation in the quota-tools
		package for more details
		(http://sourceforge.net/projects/linuxquota).
bh	(*)	ext3 associates buffer heads to data pages to
nobh		(a) cache disk block mapping information
		(b) link pages into transaction to provide
		ordering guarantees.
		"bh" option forces use of buffer heads.
		"nobh" option tries to avoid associating buffer
		heads (supported only for "writeback" mode).

Specification
==========

Ext3 shares all disk implementation with the ext2 filesystem, and adds transactions capabilities to ext2. Journaling is done by the Journaling Block Device layer.

Journaling Block Device layer

The Journaling Block Device layer (JBD) isn't ext3 specific. It was designed

to add journaling capabilities to a block device. The ext3 filesystem code will inform the JBD of modifications it is performing (called a transaction). The journal supports the transactions start and stop, and in case of a crash, the journal can replay the transactions to quickly put the partition back into a consistent state.

Handles represent a single atomic update to a filesystem. JBD can handle an external journal on a block device.

Data Mode

There are 3 different data modes:

* writeback mode
In data=writeback mode, ext3 does not journal data at all. This mode provides a similar level of journaling as that of XFS, JFS, and ReiserFS in its default mode - metadata journaling. A crash+recovery can cause incorrect data to appear in files which were written shortly before the crash. This mode will typically provide the best ext3 performance.

* ordered mode
In data=ordered mode, ext3 only officially journals metadata, but it logically groups metadata and data blocks into a single unit called a transaction. When it's time to write the new metadata out to disk, the associated data blocks are written first. In general, this mode performs slightly slower than writeback but significantly faster than journal mode.

* journal mode
data=journal mode provides full data and metadata journaling. All new data is written to the journal first, and then to its final location.
In the event of a crash, the journal can be replayed, bringing both data and metadata into a consistent state. This mode is the slowest except when data needs to be read from and written to disk at the same time where it outperforms all other modes.

Compatibility

Ext2 partitions can be easily convert to ext3, with `tune2fs -j <dev>`.
Ext3 is fully compatible with Ext2. Ext3 partitions can easily be mounted as Ext2.

External Tools
==============
See manual pages to learn more.

tune2fs: create a ext3 journal on a ext2 partition with the -j flag.
mke2fs: create a ext3 partition with the -j flag.
debugfs: ext2 and ext3 file system debugger.
ext2online: online (mounted) ext2 and ext3 filesystem resizer

References
==========

kernel source:	<file:fs/ext3/> <file:fs/jbd/>
programs:	http://e2fsprogs.sourceforge.net/ http://ext2resize.sourceforge.net
useful links:	http://www.ibm.com/developerworks/library/l-fs7.html http://www.ibm.com/developerworks/library/l-fs8.html