**Maintaining a Journal**  
Ext2fs is a traditional filesystem. Although it’s a good performer, it suffers from a major  
limitation: after a power failure, a system crash, or another uncontrolled shutdown, the  
filesystem could be in an inconsistent state. The only way to mount the filesystem safely  
so that you’re sure its data structures are valid is to perform a full disk check on it, as  
described in the next section, “Checking Filesystems.” This task is usually handled automatically when the system boots, but it takes time—probably several minutes, or perhaps  
more than an hour on a large filesystem or if the computer has many smaller filesystems.

The solution to this problem is to change to a *journaling filesystem*. Such a filesystem  
maintains a *journal*, which is a data structure that describes pending operations. Prior to  
writing data to the disk’s main data structures, Linux describes what it’s about to do in  
the journal. When the operations are complete, their entries are removed from the journal.  
Thus, at any given moment, the journal should contain a list of disk structures that *might*  
be undergoing modification. The result is that, in the event of a crash or power failure, the  
system can examine the journal and check only those data structures described within it.  
If inconsistencies are found, the system can roll back or complete the changes, returning  
the disk to a consistent state without checking every data structure in the filesystem. This  
greatly speeds the disk-check process after power failures and system crashes. Today, journaling filesystems are the standard for most Linux disk partitions. Very small partitions  
(such as a separate /boot partition, if you use one) and small removable disks (such as USB  
sticks) often lack journals, though.

Five journaling filesystems are common on Linux: ext3fs, ext4fs, ReiserFS, XFS, and  
JFS. Of these, the last three require little in the way of journal configuration. Ext3fs is a bit  
different; it’s basically just ext2fs with a journal added. This fact means you can add a journal to an ext2 filesystem, converting it into an ext3 filesystem. This is what the -j option to  
tune2fs does, as described earlier in “Adjusting Tunable Filesystem Parameters.” Ext4fs is  
a further enhancement of this filesystem family.

Adding a journal alone won’t do much good, though. To use a journal, you must mount  
the filesystem with the correct filesystem type code—ext3 rather than ext2 for ext3fs or  
ext4 for ext4fs. (The upcoming section “Mounting and Unmounting Filesystems” describes  
how to do this.)  
The journal, like other filesystem features, has its own set of parameters. You can  
set these with the -J option to tune2fs. In particular, the size=journal-size and  
device=external-journal suboptions enable you to set the journal’s size and the device on  
which it’s stored. By default, the system creates a journal that’s the right size for the filesystem and stores it on the filesystem itself.