

# OutOfFileDescriptors

There could be various reasons for this:

1 The process is running under a tight limit. See `limit/ulimit` docs for more details. 1 Your system has a tight limit. See the OS docs for info on increasing the limit. 1 Weird bugs in third-party code. For instance, from Bugzilla ticket 3326, it was found that `spamd` children would eventually run out of open file descriptors. Using `lssof` to look at the list of open files, a developer found that a certain `bayes_seen` database was remaining open, even though it was reported closed by `DB_File`'s `untie()` function:

I traced the problem down to `seen_put()` trying to put the message id into the seen database. if we try to put something in the DB, the fd stays open, even though we've `untie()`'d the DB. Did some more tracing... Best guess is that `DB_File`'s `untie()` is failing internally, and since there's no return code, we can't find that out. Here's why we guess that:

```
<felicity> $ db_verify bayes_seen
<felicity> db_verify: Page 3981: non-empty page in unused hash bucket 3333
<felicity> db_verify: Page 0: page 1273 encountered a second time on free list
<felicity> db_verify: DB->verify: bayes_seen: DB_VERIFY_BAD: Database verification failed
<felicity> db_dump bayes_seen | db_load bayes_seen.new
<felicity> mv bayes_seen.new bayes_seen
<felicity> now everything works.
```

by using the Berkeley DB tools and rebuilding the DB, the problem was cleared, and `untie()` works fine.