

Item 82

**git\_comments:**

**git\_commits:**

1. **summary:** LUCENE-5958: add more efficient test  
**message:** LUCENE-5958: add more efficient test git-svn-id:  
<https://svn.apache.org/repos/asf/lucene/dev/trunk@1625538> 13f79535-47bb-0310-9956-ffa450edef68

**github\_issues:**

**github\_issues\_comments:**

**github\_pulls:**

**github\_pulls\_comments:**

**github\_pulls\_reviews:**

**jira\_issues:**

1. **summary:** OOM or exceptions during checkpoint make IndexWriter have a bad day  
**description:** During finishCommit(), we run checkpoint after we wrote the commit to disk, but if things go wrong here (e.g. IOException when IFD deletes a pending file, OOM), then everything will go wrong (we won't even properly incref things, and may end out deleting wrong files if the user calls rollback, leaving a corrupt index).

**jira\_issues\_comments:**

1. **body:** here is a 4.x seed: ant test -Dtestcase=TestIndexWriterOutOfMemory -Dtests.method=testBasics -Dtests.seed=EEE122F7078D3AF2 -Dtests.slow=true -Dtests.locale=ro -Dtests.timezone=Asia/Ho\_Chi\_Minh -Dtests.file.encoding=ISO-8859-1 in this case, we get an exception during IFD.deleteFile (one that isnt an IOException), and it happens during deletePendingFiles...  
**label:** test
2. The thing is, the commit has already gone down, we cant throw any exception to the user here, because there is "no going back". The only good solution i have, is similar to 'hitOOM' (maybe we can generalize that?) to hold exceptions like this, and deliver them in the future instead. In such a case (hitOOM=true or whatever we rename it to), we should definitely never delete any files for any reason. Leave it to the next writer.
3. **body:** I think when things go wrong like this, its ideal for the user to get them asap (e.g. next addDocument, not commit or whatever). So I think we should close the writer (like closedbyinterruptexception in java when state gets messed up), so the next ensureOpen fails. Instead of a hitOOM-type boolean, I think we should save any exception/error that causes this, and ensureOpen passes it to ACE as the cause. In the general case it will be null.  
**label:** code-design
4. Commit 1625538 from [~rcmuir] in branch 'dev/trunk' [ <https://svn.apache.org/r1625538> ] LUCENE-5958: add more efficient test
5. Commit 1625543 from [~rcmuir] in branch 'dev/branches/branch\_4x' [ <https://svn.apache.org/r1625543> ] LUCENE-5958: add more efficient test
6. +1 for a "closed by OOM" approach, and saving the exc that "caused" the unexpected closing.
7. **body:** Here's a first stab: when we hit 'unrecoverable exception' like this, we record the exception (instead of hitOOM), close ourselves. In such a "screwed" state IndexFileDeleter also refuses to delete any files so we wont corrupt anything. This is better on the user i think, because previously they had to do this OOM handling themselves, but now IW will throw ACE (with root cause of why it had to do this) on any future operation.  
**label:** code-design
8. **body:** By the way, this patch still relays the exception in finishCOMmit (after we have actually committed!). It just prevents index corruption. I think its confusing to relay the exception too? So i think

we should suppress it in that case... otherwise the user probably thinks their commit failed when it succeeded...

**label:** code-design

9. **body:** Patch looks great, I love the "really bad day" comment. In IFD.ensureOpen, we currently check for writer == null case, but your new writer.tragedy check doesn't check for that ... but then I don't think it's possible for writer to be null: it's final, and we pass to ctor, and IFD is only ever created by IW, and IW passes "this" to it, so I think just remove the null check?  
**label:** code-design
10. Thanks Mike. I will remove that useless null check. I'll also add a comment about why ensureOpen "isnt enough" (because we pass false when we are closing, but we dont want to delete files if we are closing because of a disaster)
11. Commit 1625853 from [~rcmuir] in branch 'dev/trunk' [ <https://svn.apache.org/r1625853> ] LUCENE-5958: OOM or exceptions during checkpoint make IndexWriter have a bad day
12. Commit 1625859 from [~rcmuir] in branch 'dev/branches/branch\_4x' [ <https://svn.apache.org/r1625859> ] LUCENE-5958: OOM or exceptions during checkpoint make IndexWriter have a bad day
13. reopen for 4.10.1
14. Commit 1626189 from [~rcmuir] in branch 'dev/branches/lucene\_solr\_4\_10' [ <https://svn.apache.org/r1626189> ] LUCENE-5958: OOM or exceptions during checkpoint make IndexWriter have a bad day
15. Commit 1626363 from [~rcmuir] in branch 'dev/trunk' [ <https://svn.apache.org/r1626363> ] LUCENE-5958: add logic to merge exc handling as well
16. Commit 1626366 from [~rcmuir] in branch 'dev/branches/branch\_5x' [ <https://svn.apache.org/r1626366> ] LUCENE-5958: add logic to merge exc handling as well
17. Commit 1626368 from [~rcmuir] in branch 'dev/branches/lucene\_solr\_4\_10' [ <https://svn.apache.org/r1626368> ] LUCENE-5958: add logic to merge exc handling as well
18. Commit 1626750 from [~rcmuir] in branch 'dev/trunk' [ <https://svn.apache.org/r1626750> ] LUCENE-5958: handle illegalstateexception 'checks' by IW as well
19. Commit 1626751 from [~rcmuir] in branch 'dev/branches/branch\_5x' [ <https://svn.apache.org/r1626751> ] LUCENE-5958: handle illegalstateexception 'checks' by IW as well
20. Commit 1626752 from [~rcmuir] in branch 'dev/branches/lucene\_solr\_4\_10' [ <https://svn.apache.org/r1626752> ] LUCENE-5958: handle illegalstateexception 'checks' by IW as well
21. **body:** The changes caused a risk of deadlock in IW, which we hit in recent nightly test failure. Here's a patch to fix that; the issue is we cannot be sync'd on IW when dealing with a tragedy ... there were two places in IW where we were doing that. I added an assert that we don't hold the lock, and fixed those two places. I tried to simplify finishCommit as much as I could, and get the infoStream logging out of the tragedy handling as much as I could ...  
**label:** code-design
22. I upgraded to blocker: we need to fix this before releasing 4.10.1.
23. perfect, i reviewed all the places calling this in the patch. patch looks great, thank you.
24. Commit 1627003 from [~mikemccand] in branch 'dev/trunk' [ <https://svn.apache.org/r1627003> ] LUCENE-5958: fix IW deadlock
25. Commit 1627005 from [~mikemccand] in branch 'dev/branches/branch\_5x' [ <https://svn.apache.org/r1627005> ] LUCENE-5958: fix IW deadlock
26. Commit 1627009 from [~mikemccand] in branch 'dev/branches/lucene\_solr\_4\_10' [ <https://svn.apache.org/r1627009> ] LUCENE-5958: fix IW deadlock
27. Bulk close for Lucene/Solr 4.10.1 release