

git_comments:

1. * Returns the configured {@link DocumentsWriterPerThreadPool} instance. * @see #setIndexerThreadPool(DocumentsWriterPerThreadPool) * @return the configured {@link DocumentsWriterPerThreadPool} instance.
2. * Expert: Sets the {@link DocumentsWriterPerThreadPool} instance used by the * IndexWriter to assign thread-states to incoming indexing threads. If no * {@link DocumentsWriterPerThreadPool} is set {@link IndexWriter} will use * {@link ThreadAffinityDocumentsWriterThreadPool} with max number of * thread-states set to {@value #DEFAULT_MAX_THREAD_STATES} (see * {@link #DEFAULT_MAX_THREAD_STATES}). * </p> * <p> * NOTE: The given {@link DocumentsWriterPerThreadPool} instance must not be used with * other {@link IndexWriter} instances once it has been initialized / associated with an * {@link IndexWriter}. * </p> * <p> * NOTE: This only takes effect when IndexWriter is first created.</p>
3. * Returns the max number of simultaneous threads that may be indexing * documents at once in IndexWriter. * <p> * To modify the max number of thread-states a new * {@link DocumentsWriterPerThreadPool} must be set via * {@link #setIndexerThreadPool(DocumentsWriterPerThreadPool)}. * </p> * @see #setIndexerThreadPool(DocumentsWriterPerThreadPool)

git_commits:

1. **summary:** LUCENE-3023: cleaned up javadoc with respect to DWPT changes
message: LUCENE-3023: cleaned up javadoc with respect to DWPT changes git-svn-id: https://svn.apache.org/repos/asf/lucene/dev/branches/realtime_search@1097381 13f79535-47bb-0310-9956-ffa450edef68

github_issues:**github_issues_comments:****github_pulls:****github_pulls_comments:****github_pulls_reviews:****jira_issues:**

1. **summary:** Land DWPT on trunk
description: With LUCENE-2956 we have resolved the last remaining issue for LUCENE-2324 so we can proceed landing the DWPT development on trunk soon. I think one of the bigger issues here is to make sure that all JavaDocs for IW etc. are still correct though. I will start going through that first.
2. **summary:** Land DWPT on trunk
description: With LUCENE-2956 we have resolved the last remaining issue for LUCENE-2324 so we can proceed landing the DWPT development on trunk soon. I think one of the bigger issues here is to make sure that all JavaDocs for IW etc. are still correct though. I will start going through that first.
3. **summary:** Land DWPT on trunk
description: With LUCENE-2956 we have resolved the last remaining issue for LUCENE-2324 so we can proceed landing the DWPT development on trunk soon. I think one of the bigger issues here is to make sure that all JavaDocs for IW etc. are still correct though. I will start going through that first.
4. **summary:** Land DWPT on trunk
description: With LUCENE-2956 we have resolved the last remaining issue for LUCENE-2324 so we can proceed landing the DWPT development on trunk soon. I think one of the bigger issues here is to make sure that all JavaDocs for IW etc. are still correct though. I will start going through that first.
5. **summary:** Land DWPT on trunk
description: With LUCENE-2956 we have resolved the last remaining issue for LUCENE-2324 so we can proceed landing the DWPT development on trunk soon. I think one of the bigger issues here is to make sure that all JavaDocs for IW etc. are still correct though. I will start going through that first.

label: documentation

- [illegible]

make sure that all JavaDocs for IW etc. are still correct though. I will start going through that first.

- [illegible]

jira_issues_comments:

1. I added a [jenkins build|https://builds.apache.org/hudson/job/Lucene-Solr-tests-only-realtime_search-branch/] that runs every 4 hours to give the RT branch some exercise. I added my email address and buschmi to the recipients if the build fails.... if you wanna be added let me know. From now on we should only commit bugfixes, documentation and merges with trunk to this branch. From my point of view there is only one blocker left here (LUCENE-3028) so the remaining work is mainly reviewing the current state and polishing the javadocs. I will go over IW, IR and DW java docs as a start.
2. Why not just email dev@ when it fails? Since it will soon land I think all should feel pain when it fails ;)
3. bq. Why not just email dev@ when it fails? Since it will soon land I think all should feel pain when it fails true, done!
4. selckin thanks for reporting those failures... I fixed the TestIndexWriterException ones in LUCENE-3031 and LUCENE-3032. The TestOmitTf failure caused by a recently fixed bug on trunk (LUCENE-3027) which I haven't merged into RT branch yet. I just did the merge and that fixes that issue too. I will commit the merge in a minute The issues you are reporting with addIndexes I can not reproduce though... I will spin off issues for them.
5. **body:** Attached patch. I made a pass through the diffs of RT vs trunk, and made a bunch of small fixes (eg whitespace, typos) but there were places I had real questions so I put // nocommits in... Can we rename Healthiness -> DocumentsWriterStallControl (or something like that)? I think we lost this infoStream output from trunk? {noformat} message(" ramUsed=" + nf.format(startMBUsed) + " MB" + " newFlushedSize=" + nf.format(newSegmentSize) + " MB" + " (" + nf.format(newSegmentSizeNoStore) + " MB w/o doc stores)" + " docs/MB=" + nf.format(numDocs / newSegmentSize) + " new/old=" + nf.format(100.0 * newSegmentSizeNoStore / startMBUsed) + "%"); {noformat} It's looking good!
label: documentation
6. bq. Attached patch. awesome mike, I think you should commit that patch and we iterate once we are back from vacation? The RT hudson build tolerates //nocommit bq. I think we lost this infoStream output from trunk? do you recall where it was? bq. Can we rename Healthiness -> DocumentsWriterStallControl (or something like that)? sure go ahead can you rename the test to testStalled instead of testHealthiness bq. It's looking good! I think we are reasonably close!
7. fyi, since last night's trunk merge (r1092636) not a single test has failed in my 'while true; ant clean test' (addIndexes LUCENE-3033 has not failed since either)
8. bq. fyi, since last night's trunk merge (r1092636) not a single test has failed in my 'while true; ant clean test' (addIndexes LUCENE-3033 has not failed since either) awesome... thanks for reporting back! bq. Attached patch. mike I committed your patch to the branch so we can iterate!
9. bq. mike I committed your patch to the branch so we can iterate! I just committed some fixes and added javadoc to several classes. We are down to 0 nocommits!! Yay!
10. I was helping Simon look at reintegrating this branch (produce a patch for easy review, etc), but I found some problems. 1. it looks like some commits were marked as merged from trunk, but not actually merged. so if we reintegrate into trunk we will lose some changes. 2. some files have lost their svn:eol-style, which makes the comparison.... difficult. I'm looking at these issues now.
11. ok, i think these issues are resolved. I'm attaching the script Mike wrote that I used for checking that we don't lose any changes (I think its the same script we used for the flex branch). the way I did it is to checkout a/ and b/, reintegrate the branch into b/, and run the script to produce a huge patch. if some things look suspicious like they are "lost" changes, then i reverse apply the huge patch to the branch with eclipse and only selectively apply those lost changes and then commit.
12. Attached is the DWPT branch in patch format against trunk (for easier reviewing).
13. What about TestIndexWriter.testIndexingThenDeleting? I noticed in the branch the test method is changed to _testIndexingThenDeleting (disabled). However, if i re-enable it (rename it back) it never seems to finish...
14. {quote} I noticed in the branch the test method is changed to _testIndexingThenDeleting (disabled). However, if i re-enable it (rename it back) it never seems to finish... {quote} I just reenabled, fixed and committed that testcase on branch.
15. bq. Attached is the DWPT branch in patch format against trunk (for easier reviewing). Awesome, Thanks Robert!!!
16. I went through IW and IWC jdocs to bring them uptodate. here is a patch against the branch... Review from a native speaker would be very much welcome
17. **body:** bq. Can we rename Healthiness -> DocumentsWriterStallControl (or something like that)? I added a TODO for this - lets do that on trunk bq. I think we lost this infoStream output from trunk? just

committed a fix for this to reenable it on branch.. kind of tricky since we now flush concurrently
NumberFormat must be accessed single threaded. So I added it to DWPT#flush()

label: code-design

18. FYI - I committed the javadoc changes to branch after mikes +1 on IRC. I also marked IWC#setIndexerThreadPool as expert API
19. attached (LUCENE-3023_svndiff.patch) is just the output from 'svn diff' after merging, for reviewing property changes and similar police work :)
20. here is my first review round. I manually ported some missed merges from trunk and fixed some really minor things. I will commit shortly to branch if nobody objects
21. bq. I will commit shortly to branch if nobody objects committed... robert is running a new reintegration round now
22. I resynced up to r1097442 and here are the latest patches (the full patch and the svn diff)
23. Modified the Python script a bit to do a recursive diff b/w two dirs to make an applicable patch -- added a usage, and a -skipWhitespace option. I put it under a new 'dev-tools/scripts' dir...
24. **body:** bq. I put it under a new 'dev-tools/scripts' dir... +1 mike can you add a little doc string to the script explaining what it does and how to use it? I think we should also have a wiki page that explains how to reintegrate a branch just like we have one for merging changes into a branch.
label: documentation
25. here is my first cut at CHANGES.TXT for landing on trunk. Review would be much appreciated.
26. Iteration on diffSources.py -- adds usage line, copyright header. I think it's ready to be committed!
27. Small edits to Simon's CHANGES entry.
28. I committed the CHANGES.TXT patch to branch. I think we should freeze the branch now so robert can create a last final patch. We should let that patch linger around for a while, yet I plan to commit this to trunk on monday. Good work everybody!
29. I merged the frozen branch again. Attached is a first patch for reviewing code changes (not SVN diff), created by the following command between 2 fresh checkouts, one of them "svn merge --reintegrate":
{noformat} diff -uNrNb --strip-trailing-cr trunk-lusolr1 trunk-lusolr2 | filterdiff -x "*.svn*" --strip 1 --clean
> LUCENE-3023.patch {noformat} This patch is not intended to be applied, its more to verify the changes (therefore all whitespace changes created by merging were excluded).
30. Here is the final SVN diff. To work around some itches with SVN, the following was done: - reverted everything outside lucene sub folder - used the previously created manual diff to get a list of all changed files (using patchutils command lsdiff) - used "svn -q status | sed 's/^...../' > ../svn-files.txt" to get all files affected after merge - intersect both files (lsdiff and svn status one) to find all files that are in reality unchanged, but still affected by SVN (these are all files that were added after branching - this is a known limitation of SVN. Files added after branching are "replaced" by merge reintegrate, losing all history). Store those files in unchanged.txt - use the intersected filelist and revert everything: cat ../unchanged.txt | xargs svn revert - finally do a record-only merge again to fix mergeprops reverted by the previous revert
My checkout is now ready to commit. If we have some minor problems with the patch, please wait with fixing after commit. If there are serious problems, we can fix them in realtime branch and merge manually (I can do that later).
31. Here finally all whitespace changes in one patch. They will be committed, but are not included in the main patch.
32. Just wanted to say: you guys totally rock! Great teamwork here with all the work involved of getting the branch merged back. I'm sorry I couldn't help much in the last few weeks.
33. This looks awesome guys! I've started some ad-hoc testing via Solr. A single threaded CSV upload (bulk indexing... no real-time reopens) looks pretty much the same, and doing 2 CSV uploads at once was 36% faster (a bit apples-to-oranges since the number of resulting segments was also higher... but even still, looks like a good improvement!)
34. +1 to commit! Great work everyone :)
35. +1 to commit... I reviewed one more time - looks good :)
36. +1! (after a not-so-thorough review ;-)
37. **body:** Just a few minor comments, otherwise +1 to commit! (I'm super excited this is finally happening after the branch was created a year ago or so!) * In DocumentsWriterPerThreadPool: remove: {code} +
//public abstract void clearThreadBindings(ThreadState perThread); + + //public abstract void
clearAllThreadBindings(); + {code} * In ThreadAffinityDocumentsWriterThreadPool#getAndLock() we
had talked about switching from a per-threadstate queue (safeway model) to a single queue (whole foods).
I'm wondering if we should do that before we commit or change that later as a separate patch? * remove
some commented out code in TestFlushByRamOrCountsPolicy#testHealthiness
label: code-design

38. bq. In ThreadAffinityDocumentsWriterThreadPool#getAndLock() we had talked about switching from a per-threadstate queue (safeway model) to a single queue (whole foods). I'm wondering if we should do that before we commit or change that later as a separate patch? I will create a new issue for this I don't think this should block the commit merge! all other minor stuff can be done after committing. I will take care of these issues.
39. Thanks Simon! As noted before, this is minor and should be done after the merge was committed, else it would produce lot of useless work :(I will commit this tomorrow morning MEST. After committing, I will move the branch as a new SVN tag (realtime_DWPT_latest or like that) and if somebody wants to start again with more realtime work, we should branch again (according to the SVN book: a reintegrated branch cannot be used for development anymore and should be removed).
40. Committed merged branch to trunk revision: 1098427 Moved branch away as tag in revision: 1098428
41. bq. In ThreadAffinityDocumentsWriterThreadPool#getAndLock() we had talked about switching from a per-threadstate queue (safeway model) to a single queue (whole foods). I'm wondering if we should do that before we commit or change that later as a separate patch? I opened LUCENE-3060 for this. @buschmi maybe you can add some more info to that issue if you recall the discussion? {quote} Committed merged branch to trunk revision: 1098427 Moved branch away as tag in revision: 1098428 {quote} AWESOME! :)
42. The first full Jenkins Build also succeeded. When reviewing the first Clover Build report, I noticed 2 new final classes, that have no code coverage at all (see [<https://builds.apache.org/hudson/job/Lucene-trunk/1549/clover-report/org/apache/lucene/index/pkg-summary.html>]): - DocFieldConsumers - DocFieldConsumersPerField I am not sure if those are old relicts (dead code) or newly added ones, but not yet used.
43. I reopen this one, as the merge added a reincarnation of quicksort in DocFieldProcessor (which was previously removed in the corresponding *PerThread class, but lost during the merge). I will fix soon.
44. Here the patch. Will commit soon.
45. Removed quicksort in revision 1098592
46. **body:** This cutover to concurrent flushing (DWPT) produces astounding increases in indexing throughput: <http://people.apache.org/~mikemccand/lucenebench/indexing.html> 186 GB plain text per hour (from 101 GB/hour the day before)!!! It's not every day you see an 84% jump in indexing throughput! Wow. This is on a machine that has substantial CPU+IO concurrency, ie, it was bottlenecked by our non-concurrent flush. Also, I can now tune up the IW settings I use in those nightly benchmarks; it's now 6 threads and "only" 512 MB RAM buffer. I'll wait a few days and then do that. Looks like a few queries got a bit slower... I suspect this is because the index segment count has changed. Before concurrent flushing it was this: {noformat} 36(4.0):C4977400 _69(4.0):C4977400 _9c(4.0):C4977400 _cf(4.0):C4977400 _fi(4.0):C4977400 _fq(4.0):C497740 _g1(4.0):C497740 _gc(4.0):C497740 _gn(4.0):C497740 _gy(4.0):C497740 _gx(4.0):C49774 _gz(4.0):C49774 _h0(4.0):C49774 _h1(4.0):C49774 _h2(4.0):C49774 _h3(4.0):C468 {noformat} After concurrent flushing: {noformat} _3d(4.0):C4977400 _6h(4.0):C4977400 _9j(4.0):C4977400 _cn(4.0):C4977400 _fq(4.0):C4977400 _fu(4.0):C497740 _g6(4.0):C497740 _gh(4.0):C497740 _gs(4.0):C497740 _h2(4.0):C497740 _gy(4.0):C49774 _gz(4.0):C49774 _h0(4.0):C49774 _h5(4.0):C4105 _1(4.0):C2627 _h4(4.0):C16331 _h3(4.0):C28728 _h1(4.0):C48225 {noformat} So we have 2 extra segments... it's interesting how this affects some queries but not others.

label: code-design