Item 84
git_comments:

git_commits:

1. **summary:** LUCENE-1654: Fix UOE with old-style user-data indexes.
   message: LUCENE-1654: Fix UOE with old-style user-data indexes. git-svn-id: https://svn.apache.org/repos/asf/lucene/java/trunk@779686 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
2. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
3. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
4. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
5. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it

later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.

6. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
   label: code-design
7. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
8. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
9. **summary:** Include diagnostics per-segment when writing a new segment
   description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
   label: code-design
10. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
    label: code-design
11. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
    label: code-design
12. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not

public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.

13. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
14. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
15. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.
    label: code-design
16. **summary:** Include diagnostics per-segment when writing a new segment
    description: It would be very helpful if each segment in an index included diagnostic information, such as the current version of Lucene. EG, in LUCENE-1474 this would be very helpful to see if certain segments were written under 2.4.0. We can start with just the current version. We could also consider making this extensible, so you could provide your own arbitrary diagnostics, but SegmentInfo/s is not public so I think such an API would be "one-way" in that you'd have to use CheckIndex to check on it later. Or we could wait on such extensibility until we provide some consistent way to access per-segment details in the index.

jira_issues_comments:

1. Let's have string key-value pairs per-segment, per-commit, per-index? Lucene can store debugging data there, user can add something if he needs. We can design some unified metadata interface available through IndexReader and won't expose SegmentInfo (doing this with our current back-compat policies is akin to suicide)
2. bq. Let's have string key-value pairs per-segment, per-commit, per-index? This would be great, but it's quite a bit more work than having IW include its own diagnostics per-segment. Can you open a separate issue for that?
3. We can start with string key-value pairs per-segment that are created only from inside Lucene and leave everything else for the latter. What I'm trying to avoid is writing a partial solution, then a more generic one (with different implementation) and having to keep that partial solution around because it was already released. I.e. you're going to extend segment file format, right? Let it be key-value pairs from the start, instead of a single Int that is predetermined to be Lucene's version.
4. OK I'll start with that.
5. OK I added per-segment private storage of diagnostics within IW. It records the current Lucene version plus OS & Java version details, and the reason for the segment (flush or merge) as a Map<String, String>. I also changed the commitUserData to be a Map<String,String> too. This API has not yet been released so we are free to change it. CheckIndex now prints both the commit level and per-segment level maps. I added Constants.LUCENE_VERSION to record the current version. I try to look up the impl version from the manifest and use that, else I fallback to a String constant (now 2.9-dev). I also added a unit test

that asserts this value matches what's in common-build.xml. When I commit I'll update ReleaseTodo on the wiki to remember to update this value. I plan to commit in a day or two.

6. **body:** Let's use Collections.EMPTY_MAP instead of new HashMap() for empty maps? If I correctly read the patch, you retained HasUserData/CommitUserData in file format description, while it was totally removed from the code (except for back-compat read).
   label: code-design
7. Excellent catches Earwin, thanks! New patch attached.
8. This seems useful for including debug info?
9. **body:** bq. This seems useful for including debug info? You mean making the API public? It's the reading side of that API that makes me nervous (SegmentInfo/s is not public). Or do you think it'd be useful to allow one to "setDiagnostics(Map<String,String>)" in IW, but only see such diagnostics on running CheckIndex? That'd be a much smaller change.
   label: code-design
10. **body:** bq. It's the reading side of that API that makes me nervous (SegmentInfo/s is not public). No-no-no, don't open it up, it's suicidal :) We can theoretically allow access to per-segment data from segment readers, as they have 1-1 relation. So, when I finish always-use-MSR patch, you should be able to get per-commit data from MSR and per-segment from SR using the same API. Sounds somewhat dirty, but would work well... needs more thought.
    label: code-design
11. **body:** bq. You mean making the API public? I was thinking it could be (not sure) useful for debugging or assertions in LUCENE-1313. I wasn't sure if this was an intended use or would simply be extra noise?
    label: code-design
12. OK let's hold off on the public API. Jason can you give an example of how you'd use this in LUCENE-1313? I don't see the connection. The initial intention was exactly cases like LUCENE-1474 where I'd really like to know which Lucene version wrote a given segment, in helping to debug.
13. I found a bug for indexes with the string-based userdata: {code} if (format <= FORMAT_USER_DATA) { if (format <= FORMAT_DIAGNOSTICS) { userData = input.readStringStringMap(); } else { userData = Collections.EMPTY_MAP; if (0 != input.readByte()) { userData.put("userData", input.readString()); } } } else { userData = Collections.EMPTY_MAP; } {code} But the empty maps of Collections are read-only, so put() throws UOE. I fix this and commit: userData should be a singleton map, if 0!=readByte(), otherwise EMPTY_MAP.
14. Woops -- thanks Uwe!