Item 85
git_comments:

git_commits:

1. **summary:** LUCENE-2562: Preserve map key order for the user interface.
   message: LUCENE-2562: Preserve map key order for the user interface.

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Make Luke a Lucene/Solr Module
   description: see "RE: Luke - in need of maintainer": http://markmail.org/message/m4gsto7giltvrpuf "Web-based Luke": http://markmail.org/message/4xwps7p7ifltme5q I think it would be great if there was a version of Luke that always worked with trunk - and it would also be great if it was easier to match Luke jars with Lucene versions. While I'd like to get GWT Luke into the mix as well, I think the easiest starting point is to straight port Luke to another UI toolkit before abstracting out DTO objects that both GWT Luke and Pivot Luke could share. I've started slowly converting Luke's use of thinlet to Apache Pivot. I haven't/don't have a lot of time for this at the moment, but I've plugged away here and there over the past work or two. There is still a *lot* to do.

jira_issues_comments:

1. **body:** Re: Apache Pivot So far its been a nice experience - the main draw back to it from my perspective is that it heavily favors fixed sized layouts. I really don't like this - I like fluid, resizing layouts. So far I haven't been able to make that happen, but its early yet - though it may be more difficult than worth it short term. For now I am just targeting a decent fixed size while the bulk of the grunt work is being done.
   label: code-design
2. I haven't had any time to really work on this in a while, but I did bite the bullet and join the pivot mailing list and figured out my issues with making a fluid resizing layout - which is sweet and will hopefully motivate me to make some progress here soon.
3. So, there is really a lot to do here - but I have gotten a fair amount done. I'm near the point that I would like to share something - mostly so that I can perhaps get some help doing more. I think we can start the module well before this version of Luke can do everything the current version of Luke can - but I'd like a spot to share something that's not even really ready for a module spot. I'm looking for a shared scratch pad to use to get to a rough module point. Anyone mind if I make a spot in svn? lucene/luke, lucene/sandbox/luke or something? I'll pull any of the thinlet stuff out first. We may need to consider other issues like code grants, but I think we can at least have a contained starting point in svn before we get those details resolved? Thoughts? Otherwise I'll need to share on an svn outside apache, which may cause the need for further code grants issues, etc (developing outside of apache) that I'd rather just avoid. But either way. Also, if anyone is into translating, I'm working on localization from the ground up, as well as modularizing and cleaning up a lot of the code. Luke in German anyone?
4. bq. Anyone mind if I make a spot in svn? lucene/luke, lucene/sandbox/luke or something? +1 for lucene/sandbox/luke We can always move it later
5. bq. Anyone mind if I make a spot in svn? lucene/luke, lucene/sandbox/luke or something? I just wonder why you don't branch and put it already in the right location. I think we should use branches more often though. bq. Also, if anyone is into translating, I'm working on localization from the ground up, as well as modularizing and cleaning up a lot of the code. Luke in German anyone? Heh, sure I will do the German translation. - robert what about Thai :D
6. +1. I'm happy to do a code grant if people think it's necessary - after all, the non-thinlet parts are already licensed under ASL.
7. **body:** bq. I just wonder why you don't branch and put it already in the right location. I think we should use branches more often though Because I really have no use/want for a branch of lucene/solr - luke is currently completely external to that, so what's the point really? When it's in module form, perhaps that would make more sense, but then I'm prob inclined to just put it in trunk and iterate. bq. I'm happy to do a code grant if people think it's necessary - after all, the non-thinlet parts are already licensed under ASL. I think the rule is that if it was developed outside of Apache, they want the grant, regardless of the license - especially for a codebase so large. I've heard different things when the dev is a committer - but they tend to err on the side of caution. We will see what Grant says.
   label: code-design
8. I've checked in the current state at: https://svn.apache.org/repos/asf/lucene/sandbox/luke The main class is currently: org.apache.lucene.luke.ui.LukeApplication
9. If you just want to check out the current state of things, you can launch a jnlp version at http://www.lucene-solr.com/luke.jnlp
10. I've taken Mark Harwood's cool Analyzers Tool plugin and ported it over as a first class citizen with it's own tab. Great tool for checking out analysis - I'd hadn't really paid much attention to it till Robert pointed it out. Check it out - I've got it in the latest jnlp at http://www.lucene-solr.com/luke.jnlp
11. Would like to get going on pushing this in as a module - anyone want to comment on the legal aspect? Do we need a code grant or not? The original authors (AB and a couple plugin authors?) are committers, but the code was original developed outside Apache... Anyone want to weigh in so we can move forward on module integration? Robert says he might lend a hand there.
12. (P.S. Not to say this is that close to being done - would just like to get the legal stuff out of the way before spreading around too much in svn)
13. {quote} Would like to get going on pushing this in as a module - anyone want to comment on the legal aspect? Do we need a code grant or not? The original authors (AB and a couple plugin authors?) are committers, but the code was original developed outside Apache... {quote} I am not a legal expert but shooting an email to legal-discuss would bring clarification and we are on the safe side. Even if we would need one since all are around that should not be a big deal. simon
14. How many contributors have there been to Luke outside of the ASF? I know Andrzej and Mark. I know I have put up a patch, I think. Was explicit permission given to use? I guess it comes down to ownership. I suppose you could split out where those patches were applied to, as well. Given that we are dropping the Thinlet stuff, anything that was changed there doesn't matter.
15. For the record, I'm obviously happy to grant any of my Luke contributions to Apache. BTW - If anyone is looking at a revision of the GWT front end then this recent Google acquisition looks like it might be interesting: http://code.google.com/webtoolkit/tools/gwtdesigner/index.html
16. bq. shooting an email to legal-discuss I'll go to legal discuss if I have to - but we have dealt with this type of thing before - so if the right people respond, I'd rather just go that route.
17. bq. I'll go to legal discuss if I have to - but we have dealt with this type of thing before - so if the right people respond, I'd rather just go that route. What's happening with this one? It would be great to get it tied into the build in trunk.

18. Still going - just kind of a weekend, nights project, of which I've been short of recently. Last weekend, 3 day hiking camping trip, previous 2 weekends before that (and all days in between) in Boston. Looking a little wavy going forward too, but I'm still heavily invested. If Grant is not worried about getting a code grant, than neither am I. IMO, since the contributors are committers with CLA's, there is not much to worry about. Since unless someone says differently, I'm inclined to just move this (it's already in svn, so moving it under trunk is not much different). If someone has a concern, we can hit legal-discuss though.
19. {quote} Still going - just kind of a weekend, nights project, of which I've been short of recently. Last weekend, 3 day hiking camping trip, previous 2 weekends before that (and all days in between) in Boston. Looking a little wavy going forward too, but I'm still heavily invested. {quote} Right, i meant really any legal issue/moving to trunk. Though maybe it doesnt have the complete feature set yet, in trunk it would stay current and what you have so far is useful. bq. Since unless someone says differently, I'm inclined to just move this (it's already in svn, so moving it under trunk is not much different). +1
20. mark, I marked this as gsoc2011 so if somebody is interested in working on this during GSoC can apply for it. Hope that is ok for you though.
21. Yeah, that would be great! Either way, I have def not abandoned this - just don't know when it will get more love...
22. Any update on where this is? I saw something working reasonably well before: http://www.lucene-solr.com/luke.jnlp What about putting the almost working code in /trunk/modules/luke? And adding proper warnings for it being in progress We can iterate from there, but it would be awesome to have a version of luke that works with trunk!
23. bq. http://www.lucene-solr.com/luke.jnlp whoops - was not paying attention, and looks like I let that domain expire. 100 bucks to redeem it - I think I'll let her go for now :) This version is in svn somewhere under sandbox or something. Robert has threatened to move it to a module while we where at revolution :)
24. bq. This version is in svn somewhere under sandbox or something. Robert has threatened to move it to a module while we where at revolution +1
25. Where does this issue stand? I noticed a thread on the user list today, where people were practically begging for a Luke version that works with recent releases. Is there consensus to have Luke as a Lucene module? If so, is it just a legal thing, or a "forgotten issue"?
26. If people don't have time to work on this, perhaps still someone should put up a patch that doesn't compile of all Luke's code? I will be happy to take a look, and I'm sure others will too.
27. I did all the initial work on it. I haven't had even an ounce of time I could put towards it for a long time now. I had a fair amount of functionality working nicely. It was a fun little project, there are no legal things, and hopefully some day it's completed by someone :)
28. I think Mark's latest state is committed to a branch at https://svn.apache.org/repos/asf/lucene/sandbox/luke/ ? Would love to see this revived as an actual module of lucene.
29. Ah I missed that comment! indeed there's a version under sandbox/luke.
30. **body:** The work was really not that difficult. More just marching through each piece and updating to some more recent apis. I've got so many other things on my plate, even though I enjoyed this work, it's just super low priority. But I could probably help point someone in the right direction if they wanted to get started - perhaps some day I'd even help out with some more work on it when I don't feel so buried.
    label: code-design
31. Commit 1502987 from [~markrmiller@gmail.com] [https://svn.apache.org/r1502987] LUCENE-2562: hack and slash update to Lucene/Solr 4.3.1
32. I've done a quick hack and slash update to Lucene/Solr 4.3 and uploaded some screen shots of the latest state of this. To run it, simply check out https://svn.apache.org/repos/asf/lucene/sandbox/luke and run the main class org.apache.lucene.luke.ui.LukeApplication Currently, it's a full eclipse project so there is no build step or ant script or anything.
33. **body:** It's been a long time, but the last time I talked to [~rcmuir] about this, I think we came to the conclusion that the best way forward would be to tie up these basic features and perhaps stop worrying about porting over anything else from GetOpt Luke - like the plugins for instance. This already has the majority of Luke functionality implemented - it might make sense to simply polish it off and button it up, get it in as a module and then lets new features and development happen organically.
    label: code-design
34. Commit 1502989 from [~markrmiller@gmail.com] [https://svn.apache.org/r1502989] LUCENE-2562: Update build file
35. Commit 1503004 from [~markrmiller@gmail.com] [https://svn.apache.org/r1503004] LUCENE-2562: Update to Apache Pivot 2.0.2
36. Commit 1516057 from [~markrmiller@gmail.com] [https://svn.apache.org/r1516057] LUCENE-2562: update to Lucene/Solr 4.4 and Pivot 2.0.2
37. At Ajay Bhat's request, I have updated to Lucene/Solr 4.4 and Pivot 2.0.2. [~ajay bhat], feel free to comment on this JIRA issue regarding your current progress. Also, it would be great to commit what you have so far when you are ready.
38. Re:[~markrmiller@gmail.com] and everyone else, I've done a very small patch, but I'm still facing some problems. I've checked the Analyzers in the tool manually, and these Analyzers give the java.lang.InstantiationException when used. org.apache.lucene.analysis.miscellaneous.LimitTokenCountAnalyzer org.apache.lucene.analysis.miscellaneous.PatternAnalyzer org.apache.lucene.analysis.query.QueryAutoStopWordAnalyzer org.apache.lucene.analysis.snowball.SnowballAnalyzer org.apache.lucene.collation.CollationKeyAnalyzer org.apache.lucene.solr.analysis.TokenizerChain I'm a bit confused because as of Lucene 4.0alpha, the PatternAnalyzer is deprecated from misc package, but it's still present here in miscellaneous package in Lucene 4.4 release.
39. **body:** bq: I'm a bit confused because as of Lucene 4.0alpha, the PatternAnalyzer is deprecated from misc package, but it's still present here in miscellaneous package in Lucene 4.4 release. Deprecated code is maintained through one major version. So it'll stay marked as deprecate through all of 4.x and be eligible to be removed in 5.0. That way people have time to adjust...
    label: code-design
40. [~ajay bhat], can you explain the patch? It seems to only add a tokenstream reset call? Is that the only change so far? I'll take a look at the analyzer init issue as soon as I get a chance.
41. The TokenStream reset call was needed to display the tokens generated by the Analyzer. I think that's the only change that was required. The main problem for me is the analyzers above are not giving the result, which I've been looking into. I had figured that since PatternAnalyzer was deprecated it would not give the result and so it might be a good idea to remove it from the list of analyzers. But there are also some analyzers that aren't deprecated, like the Snowball Analyzer and QueryAutoStopWordAnalyzer. Also, as per the schedule of my proposal I've done some work on the themes of the Application. I'll contribute another patch for that soon.
42. LUCENE-2562.patch Date : 13th Sept, 2013 Added: Support for 5 themes, through a recursive style change function Exit option in File menu Status bar Analyzer tokenstream reset call Documentation for above features Pending Work to be done: Status Bar implementation not complete Themes cannot be applied to one tab
43. Nice, thanks Ajay! Unfortunately, I am having trouble with the patch - it won't cleanly apply to AnalyzersTab.java. I left that file as is and checked out the color scheming though - looks good!
44. One comment - I think I like the gray theme for the default best. Once I can apply a clean patch, I'll commit your current progress.
45. Thanks for the comments, [~markrmiller@gmail.com]. The only change for AnalyzersTab.java is the tokenstream reset call. I used the default theme as per the one used in original Thinlet Luke. If you'd like, I'll do a slight modification so Gray is the default theme.
46. Commit 1523019 from [~markrmiller@gmail.com] [https://svn.apache.org/r1523019] LUCENE-2562: Ajay Bhat Support for 5 themes, through a recursive style change function Exit option in File menu Status bar Analyzer tokenstream reset call Documentation for above features
47. **body:** FYI, I see a lot of the following type thing in the console: "borderColor" is not a valid style for org.apache.pivot.wtk.TextArea "activeBackgroundColor" is not a valid style for org.apache.pivot.wtk.TextArea Duplicate listener org.apache.lucene.luke.ui.LukeWindow$2@4a9a1ac added to org.apache.pivot.wtk.Component$ComponentMouseListenerList [org.apache.pivot.wtk.skin.terra.TerraPushButtonSkin@451dfada, org.apache.lucene.luke.ui.LukeWindow$2@4a9a1ac] Perhaps that is currently expected, but FYI.
    label: code-design
48. **body:** Re: ["borderColor" is not a valid style for org.apache.pivot.wtk.TextArea "activeBackgroundColor" is not a valid style for org.apache.pivot.wtk.TextArea] Duly noted. I'll make sure to look into it and take care of it in the next patch.

label: code-design

49. Hi, is this issue still active? I've attached a patch (LUCENE-2562-ivy.patch) to follow latest Lucene updates. This patche does - add ant task "ivy-resolve" and "ivy-clean" to resolve dependencies using Ivy "ivy-resolve" task creates lib-ivy directory and download necessary libraries from maven repository. - modify "compile", "dist", and other related tasks use libraries downloaded by "ivy-resolve" - make small source code changes to go with Lucene 4.10.3 lib/*.jar, lib/solr/*.jar, lib/hadopp/*.jar are no longer used for compile and distribute, so they can be removed from the repository later, except for lib/js.jar (I could not identify this.) Note: Eclipse's .classpath file is not changed. Compiled o.a.l.luke.ui.LukeApplication can browse indices generated by Lucene 4.10.3. If you're interested, I hope to take more time for the project.
50. Hey [~Tomoko Uchida] - I still think this is an important issue, I just don't have the time to drive it currently. If you have the desire to do some work on it, I'm sure I can ramp up enough to apply some patches for you. I'll try and take a look at the attached patch before long.
51. Hi Mark, Thank you for the response! My patch needs ivy jar in lib/tools directory. (I've forgot to note that.) That does not introduce additional Java codes, but brings dependency resolution. I've slowly started to catch up and play with (and modify) the codes... so I would like to create another patches to make progress.
52. Just a thought here - would it make sense to expose the luke functionality via the admin UI - use the Admin UI as the front end for luke and have it available to everyone right away? Perhaps the reworked UI in AngularJS that I am working on might make this easier (see SOLR-5507).
53. Just my 2 cents - i would like to see this as a Lucene module, not a SOLR one. Not everyone uses SOLR to use Lucene, so porting it to a SOLR only module would be no benefit in using lucene. Would be nice to still have a clean small solution to access lucene like luke does it now, not to be forced to use solr.
54. Agreed. This about making Luke a Lucene module, not about rewriting Luke as something else.
55. bq. My patch needs ivy jar in lib/tools directory. (I've forgot to note that.) [~Tomoko Uchida]: We already use IVY in Lucene/Solr. Please use the framework that is already there through common-build.xml / module-build.xml. It is enough to place ivy.xml in the module directory and make the module include the module-build.xml file.
56. Absolutely, fair enough. Exposing it through Solr UI makes sense, restricting it does not.
57. Thanks everyone for all comments, I recognize that the goal of this project is "making Luke a Lucene module". I think it will be nice if there is a REST API server for Admin UIs, and have some kind of pluggable architecture for external modules (like custom analyzers provided by Solr, Elsticsearch, etc.) But at first, I'd like to make it cover standalone Luke application features under Apache license. Dear Uwe and all, > We already use IVY in Lucene/Solr. Please use the framework that is already there through common-build.xml / module-build.xml. It is enough to place ivy.xml in the module directory and make the module include the module-build.xml file. I wanted to use the Lucene's framework, but currently, this module is placed under http://svn.apache.org/repos/asf/lucene/sandbox/ , not http://svn.apache.org/repos/asf/lucene/dev/trunk ivy.xml and ant tasks in my patch should be only temporary (for convenience in development,) until this module will be merged to trunk. So what can I do to smoothly integrate this module into Lucene trunk in the future. If there are better ways, please let me know... Thanks
58. Patch updated. I've modified Overview tab only. Progress and Status : - Missing values in upper panel (index info) were all filled. - Fields table are now sortable by field name and term counts. Pending tasks to be done: - Decoders (last pending task for Overview tab) I'm trying for decoders. It might need some sort of pluggable design (I believe Solr's decoders should be plugged, not built-in feature.) Suggestions / ideas welcome.
59. Commit 1655103 from [~markrmiller@gmail.com] [https://svn.apache.org/r1655103] LUCENE-2562: Add some Ivy support, support Lucene, Solr 4.10.3, Missing values in upper panel (index info) were all filled, Fields table are now sortable by field name and term counts. Tomoko Uchida.
60. Patch updated. I've modified Overview and Documents tab. Progress and Status: - (Overview tab) Decoders support in fields table. They are now selectable. - (Overview tab) 'more options' support in top ranking terms table except for 'Show all term docs' (That will be supported after Search tab is completed.) - (Documents tab) Layouts arrange. (I hope that's OK.) - (Documents tab) 'Browse by term' (upper center pane) fixed. - (Documents tab) Documents viewer (lower pane) fixed and support DocValues info (original Luke does not support this because docvalues are new, of course). - Decoders are now pluggable. DecoderLoader interface added. - Some fixes, refactoring Pending tasks to be done: - (Documents tab) 'Browse documents with this term' (upper right pane) is not yet fixed. It works but be buggy. I've added DecoderLoader interface and implemented SolrDecoderLoader class to make Luke work with or without Solr jars. I think we will be able to completely separate SolrDecoderLoader class from Luke's package later (or integrate this in Luke again if that is proper way.) With this patch, if you want to use Luke with Solr's Decoder (wrapper for schema.FieldType,) you need to add a JVM option "luke.ext.decoder.loader" as below when start Luke. {code} $ java -cp "build" -Dluke.ext.decoder.loader=org.apache.lucene.luke.ext.SolrDecoderLoader org.apache.lucene.luke.ui.LukeApplication {code} My approach is not good, please let me know. For SolrDecoder, need to modify ivy.xml as follows. (I could not include this modification in my patch, because 'svn diff' complains 'Cannot display: file marked as a binary type.') {code:xml} <!-- apache solr --> <dependency org="org.apache.solr" name="solr-core" rev="4.10.3" transitive="false" // remove this line conf="solr->*,!sources,!javadoc"/> {code} I add some resources to 'LukeApplication_en.json', but I'm not a fluent English speaker, so please check and correct if there's any mistakes...
61. Awesome. I'll take a look soon.
62. Patch updated. Ive modified Documents tab. (A screen shot, "lukeALE-documents.png" is attached.) Status and progress: * support 'Browse documents with this term' and 'Show positions and Offsets' * support right click options for document's fields (Show term vector, Full text, Set norm, Save field data) Pending tasks to be done (I'm going to implement them): * 'Add','Edit','Delete' operations for documents * 'Show all docs','More like this' (will be supported when Search tab completed) I understand this is not a rush job, many tasks still remain... I'd like to keep going and log updates here. Because components are well separated, I think anyone can join. :)
63. Great - been fairly busy lately, but I'll keep committing your work. Ping me with a reminder if I slack off too much.
64. I know it's homage to the original Luke default color, but we should really change the default background color :)
65. Progress and status: Honestly there are no progress in development, I'd like to suggest about merge of this Jira issue and the Luke project now on Github. The (one of) Luke project's goal, "To port the thinlet UI to an ASL compliant license framework so that it can be contributed back to Apache Lucene.", is perfectly match the goal of this issue, I hope it would be desirable way. https://github.com/DmitryKey/luke/tree/master After a short discussion at solr mailing list, Dmitry Kan, the maintainer of Luke, has created "pivot-luke" branch in luke repository. And I have merged all the latest pivot's version codes plugged in SVN to the branch. The discussion is here: http://mail-archives.apache.org/mod_mbox/lucene-solr-user/201502.mbox/%3CCAHpHuj%3DZArPSbSLp2_d1X7OD_2V0NeBsDSRVizgTDTnBiAZPMQ%40mail.gmail.com%3E The branch for pivot's version is here: https://github.com/DmitryKey/luke/tree/pivot-luke It now can be built by maven, you can check out and test it. We'll contribute back to Lucene it, I hope this is in not so distant future. Of course, developers are welcome. ---- > I know it's homage to the original Luke default color, but we should really change the default background color I promise that, gray theme is also my favorite one! :) I'll keep watching this ticket, any comments and suggestions are welcome. Thanks.
66. It's been a while... but I have ported Luke UI from Thinlet to JavaFX (I feel JavaFX is now better choice than other 3rd party Frameworks, what do you think?) JavaFX version can be downloaded here: [https://github.com/DmitryKey/luke/releases/tag/luke-7.3.1] * Licensed under ALv2 * Ported major luke functionalities (browsing index, searching docs, optimizing/checking index, etc) * Supports recent Lucene features (DocValues, PointValues, CustomAnalyzers, etc.) * Almost full scratched to utilize modern Java syntax/features and unit tested with Lucene Test Framework The GitHub repository earns close to 900 stargagers! It's good time to contribute back to Lucene project but I am not sure what is the proper way. Let me know where should I start. Anyway, please check it out and I want to hear your feedback!
67. I'll attach some screen shots to share what new UI looks like. !luke-javafx1.png|width=50%, height=50%! !luke-javafx2.png|width=50%, height=50%! !luke-javafx3.png|width=50%, height=50%!
68. Awesome, that looks great!
69. Thank you [~markrmiller@gmail.com] for response. First I need some research for Lucene build system and try to fit the Github repository (built by maven) to it... any advise is welcome.
70. Created a pull request on GitHub. Could you check it or is it better if I add a patch to the issue? https://github.com/apache/lucene-solr/pull/420
71. **body:** [~Tomoko Uchida]: thanks for the pull request. I pulled it into a branch named {{jira/lucene-2562}}, along with a few fixes and additions, mostly to get {{ant precommit}} to succeed, which it does not yet do. I haven't tried to run the code yet :). A few things I noticed: # {{validate-

source-patterns}} complains about non-static-final loggers: {noformat} validate-source-patterns: [source-patterns] invalid logging pattern [not private static final, uses static class name]: lucene/luke/src/java/org/apache/lucene/luke/models/commits/CommitsImpl.java [source-patterns] invalid logging pattern [not private static final, uses static class name]: lucene/luke/src/java/org/apache/lucene/luke/models/documents/DocumentsImpl.java [source-patterns] invalid logging pattern [not private static final, uses static class name]: lucene/luke/src/java/org/apache/lucene/luke/models/search/SearchImpl.java {noformat} I fixed a few others like these, but ^^ have ctors that take a logger, apparently just to be able to turn logging off under testing. Is this really necessary? Right now I have a nocommit that allows {{ant precommit}} to ignore these, but it needs to be resolved. # It should be possible to run Luke from a source checkout, but it's not now. # I think Luke should be packaged with all other artifacts in Lucene's binary packages, but currently {{ant package-tgz}} and {{-zip}} don't include everything (e.g. scripts under {{bin/}}); instead, the build creates a {{.zip}} file that contains a copy of all the dependencies and the {{bin/}} scripts. I think we can get rid of this Luke-only distribution, in favor of the Lucene distribution? I don't like the idea of including the Luke-with-all-dependencies {{.zip}} in the Lucene distribution, since it will include two copies of many modules' jars. # forbidden-apis is very unhappy about classes in {{javafx.*}} packages, e.g.: {noformat} [forbidden-apis] Scanning classes for violations... [forbidden-apis] Forbidden class/interface use: javafx.beans.property.BooleanProperty [non-portable or internal runtime class] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dto.search.SelectedField (SelectedField.java, field declaration of 'selected') [forbidden-apis] Forbidden class/interface use: javafx.beans.property.SimpleBooleanProperty [non-portable or internal runtime class] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dto.search.SelectedField (SelectedField.java:29) [forbidden-apis] Forbidden class/interface use: javafx.beans.property.SimpleBooleanProperty [non-portable or internal runtime class] [...] {noformat} [~thetaphi]: do you understand what ^^ is about?
label: code-design

72. **body:** Hi, the forbiddenapis problems are caused by the way how javafx was exposed in previous versions of the JDK. With JDK8 it was shipped inside rt.jar, but on other versions it is not included at all and in Java 10 it's now part of module system. The portable runtime stuff only allows classes from the well known java packages, which does not yet include {{javafx.**}}: https://github.com/policeman-tools/forbidden-apis/blob/master/src/main/java/de/thetaphi/forbiddenapis/AsmUtils.java#L39-L40 The problem JavaFX is still not an official part of the Java SE platform, it's still optional! In addition this needs to be some special exceptional check, as javafx is only valid as runtime class in Java 8, in Java 7 it's not indluded. If You run forbiddenapis with Java 10, it should pass. The reason is: Although it's shipped with the JDK and the class name is not in the above pattern, but because it's an external module - it works as expected (I have not verified this, maybe you can do it). The reason is the module name, it's external. With Java 8 we have no module names, so the check can only use the above whitelist of package names. In short: That's a limitation of forbiddenapis and my ignorance for JavaFX. I'd suggest to redefine the forbiddenapis target in the luke module to not use the "jdk-non-portable" signature and instead the more static "jdk-internal" signature (maybe thorugh an ANT property). Nevertheless, as described above, JavaFX is a "non-portable" part of the JDK as it's not part of the spec. Not sure what the best way to solve this. I'd go for the workaround now and use "jdk-internal" instead of "jdk-non-portable" for the luke module (not for others, please).
label: code-design
73. On top of that: Starting with Java 11, JavaFX is no longer shipped with the JDK, so forbiddenapis is correct: it's not portable. See https://www.infoworld.com/article/3261066/java/javafx-will-be-removed-from-the-java-jdk.html
74. Thanks [~thetaphi], I put in an Ant property to control it. That reveals a lot forbidden-api violations left to tackle now: {noformat} -check-forbidden-all: [forbidden-apis] Reading bundled API signatures: jdk-unsafe-1.8 [forbidden-apis] Reading bundled API signatures: jdk-deprecated-1.8 [forbidden-apis] Reading bundled API signatures: jdk-internal-1.8 [forbidden-apis] Reading bundled API signatures: jdk-reflection [forbidden-apis] Reading API signatures: /Users/sarowe/git/lucene-solr/lucene/tools/forbiddenApis/base.txt [forbidden-apis] Reading API signatures: /Users/sarowe/git/lucene-solr/lucene/tools/forbiddenApis/lucene.txt [forbidden-apis] Loading classes to check... [forbidden-apis] Scanning classes for violations... [forbidden-apis] Forbidden method invocation: java.util.Locale#toString() [use Locale#toLanguageTag() for a standardized BCP47 locale name] [forbidden-apis] in org.apache.lucene.luke.app.controllers.fragments.search.QueryParserController (QueryParserController.java:159) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.overview.OverviewImpl (OverviewImpl.java:159) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.overview.OverviewImpl (OverviewImpl.java:164) [forbidden-apis] Forbidden method invocation: java.lang.String#getBytes() [Uses default charset] [forbidden-apis] in org.apache.lucene.luke.models.analysis.TestAnalysisImpl (TestAnalysisImpl.java:93) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dto.overview.TermCount (TermCount.java:29) [forbidden-apis] Forbidden method invocation: java.io.PrintStream#<init>(java.io.OutputStream,boolean) [Uses default charset] [forbidden-apis] in org.apache.lucene.luke.app.util.TextAreaPrintStream (TextAreaPrintStream.java:38) [forbidden-apis] Forbidden method invocation: java.lang.String#getBytes() [Uses default charset] [forbidden-apis] in org.apache.lucene.luke.app.util.TextAreaPrintStream (TextAreaPrintStream.java:49) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:235) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:236) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:237) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:238) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:239) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:240) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:241) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:242) [forbidden-apis] Forbidden method invocation: java.util.Locale#toString() [use Locale#toLanguageTag() for a standardized BCP47 locale name] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:243) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:243) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:244) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.QueryParserConfig (QueryParserConfig.java:245) [forbidden-apis] Forbidden method invocation: java.lang.String#toLowerCase() [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.LukeController$ColorTheme (LukeController.java:302) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.LukeController$ColorTheme (LukeController.java:302) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.OverviewController (OverviewController.java:210) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in

```
org.apache.lucene.luke.app.controllers.OverviewController (OverviewController.java:212) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.DocumentsController (DocumentsController.java:225) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.DocumentsController (DocumentsController.java:302) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.DocumentsController (DocumentsController.java:325) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.DocumentsController (DocumentsController.java:349) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:110) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:123) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:134) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:151) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:169) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:187) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.commits.CommitsImpl (CommitsImpl.java:204) [forbidden-apis] Forbidden method invocation:
java.util.concurrent.Executors#newSingleThreadExecutor() [Spawns threads with vague names; use a custom thread factory (Lucene's
NamedThreadFactory, Solr's DefaultSolrThreadFactory) and name threads so that you can tell (by its name) which executor it is associated with]
[forbidden-apis] in org.apache.lucene.luke.app.controllers.dialog.menubar.CheckIndexController (CheckIndexController.java:120) [forbidden-apis]
Forbidden method invocation: java.util.concurrent.Executors#newSingleThreadExecutor() [Spawns threads with vague names; use a custom thread
factory (Lucene's NamedThreadFactory, Solr's DefaultSolrThreadFactory) and name threads so that you can tell (by its name) which executor it is
associated with] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dialog.menubar.CheckIndexController (CheckIndexController.java:184)
[forbidden-apis] Forbidden method invocation: java.text.MessageFormat#format(java.lang.String,java.lang.Object[]) [Uses default locale]
[forbidden-apis] in org.apache.lucene.luke.app.util.MessageUtils (MessageUtils.java:41) [forbidden-apis] Forbidden method invocation:
java.lang.String#<init>(byte[]) [Uses default charset] [forbidden-apis] in org.apache.lucene.luke.app.util.TextAreaAppender
(TextAreaAppender.java:57) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses
default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dto.documents.TermPosting (TermPosting.java:32) [forbidden-apis]
Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:111) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:149) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:176) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:205) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:233) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:256) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:284) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:302) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:311) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.documents.DocumentsImpl (DocumentsImpl.java:320) [forbidden-apis] Forbidden class/interface use: java.io.File
[Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.models.commits.File (File.java:30) [forbidden-apis] Forbidden class/interface use:
java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.models.commits.File (File.java:31) [forbidden-apis] Forbidden method
invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.dialog.search.ExplanationController (ExplanationController.java:74) [forbidden-apis] Forbidden
class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.AnalysisController
(AnalysisController.java:159) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.AnalysisController (AnalysisController.java:164) [forbidden-apis] Forbidden class/interface use: java.io.File
[Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.AnalysisController (AnalysisController.java:165) [forbidden-apis]
Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.AnalysisController
(AnalysisController.java:165) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.AnalysisController (AnalysisController.java:172) [forbidden-apis] Forbidden class/interface use: java.io.File
[Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.AnalysisController (AnalysisController.java:178) [forbidden-apis]
Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.AnalysisController
(AnalysisController.java:178) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses
default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.AnalysisController (AnalysisController.java:190) [forbidden-apis]
Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.app.controllers.dto.documents.TermVector (TermVector.java:41) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:150) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:185) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:224) [forbidden-apis] Forbidden method invocation:
java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in
org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:236) [forbidden-apis] Forbidden method invocation:
java.util.Arrays#copyOfRange(java.lang.Object[],int,int) [Prefer using ArrayUtil as Arrays#copyOfRange fills zeros for bad bounds] [forbidden-
apis] in org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:326) [forbidden-apis] Forbidden method invocation:
```

java.util.Arrays#copyOfRange(java.lang.Object[],int,int) [Prefer using ArrayUtil as Arrays#copyOfRange fills zeros for bad bounds] [forbidden-apis] in org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:356) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:424) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.SearchImpl (SearchImpl.java:452) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.analysis.AnalysisImpl (AnalysisImpl.java:85) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.analysis.AnalysisImpl (AnalysisImpl.java:189) [forbidden-apis] Forbidden method invocation: java.util.concurrent.Executors#newSingleThreadExecutor() [Spawns threads with vague names; use a custom thread factory (Lucene's NamedThreadFactory, Solr's DefaultSolrThreadFactory) and name threads so that you can tell (by its name) which executor it is associated with] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dialog.menubar.OptimizeController (OptimizeController.java:119) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dialog.menubar.OpenIndexController (OpenIndexController.java:155) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dialog.menubar.OpenIndexController (OpenIndexController.java:160) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.dialog.menubar.OpenIndexController (OpenIndexController.java:161) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.desktop.PreferencesImpl (PreferencesImpl.java:51) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.desktop.PreferencesImpl (PreferencesImpl.java, method declaration of 'iniFile()') [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.desktop.PreferencesImpl (PreferencesImpl.java:140) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.desktop.PreferencesImpl (PreferencesImpl.java:34) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.SearchController (SearchController.java:291) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.app.controllers.SearchController (SearchController.java:435) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.SimilarityConfig (SimilarityConfig.java:93) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.SimilarityConfig (SimilarityConfig.java:94) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.SimilarityConfig (SimilarityConfig.java:95) [forbidden-apis] Forbidden method invocation: java.lang.String#format(java.lang.String,java.lang.Object[]) [Uses default locale] [forbidden-apis] in org.apache.lucene.luke.models.search.SimilarityConfig (SimilarityConfig.java:96) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.LogsController (LogsController.java:29) [forbidden-apis] Forbidden class/interface use: java.io.File [Use NIO.2 instead] [forbidden-apis] in org.apache.lucene.luke.app.controllers.LogsController (LogsController.java:29) [forbidden-apis] Scanned 161 class file(s) for forbidden API invocations (in 0.44s), 87 error(s). {noformat}

75. Thanks, looks good in your commit. I am still unhappy that Luke will stop working with Java 11 (e.g., which will be the default after September's Update of Ubuntu 18.04).
76. Thank you for the comments and detailed explanation. > On top of that: Starting with Java 11, JavaFX is no longer shipped with the JDK, so forbiddenapis is correct: it's not portable. I have not caught up with it. I cannot find out better solutions yet, but I'll try to look for alternative UI framework that can go with future JDK updates. Replacing UIs should be possible because core luke functionalities and UI components are well separated (I hope that,) though some extra work is needed. If you have any suggestions, please let me know. :)
77. I also found this issue related to the java 11 changes: https://github.com/javafxports/openjdk-jfx/issues/52 Of course there are licensing aspects to think of too?
78. We cannot link to GPL so this is a bummer. I don't think JavaFX, unless shipped with the JDK is a viable option for Apache projects. I think we should ask ASF legal how to proceed?
79. As you know, in the ASF legal page, GNU GPL including GNU Classpath exceptions is listed in the Category X list (honestly, I read this terms yesterday and not prepared to handle such legal matters.) [https://www.apache.org/legal/resolved.html#category-x] To not to complicate matters I'd like to withdraw the merge request for now. (Thanks for all your comments so far!) While contributing back Luke to Apache Lucene project is the our long-term goal, we can continually deliver Luke via current GitHub project. For keeping Luke source codes compatible with ALv2, Self-contained application package is the one of feasible options for users they do not (or cannot) install JavaFX, in my first impression. [https://docs.oracle.com/javase/8/docs/technotes/guides/deploy/self-contained-packaging.html] Of course discussions and suggestions are appreciated!
80. **body:** Let me add one more thing, I'm afraid that I cannot ask to and negotiate with ASF on my own about complicated matters with my limited knowledge about licensing and English vocabulary. But if you think there are chances to proceed, please guide me :)
    label: code-design
81. bq. We cannot link to GPL so this is a bummer. I don't think JavaFX, unless shipped with the JDK is a viable option for Apache projects. I think we should ask ASF legal how to proceed? bq. As you know, in the ASF legal page, GNU GPL including GNU Classpath exceptions is listed in the Category X list (honestly, I read this terms yesterday and not prepared to handle such legal matters.) https://www.apache.org/legal/resolved.html#category-x bq. I'm afraid that I cannot ask to and negotiate with ASF on my own about complicated matters with my limited knowledge about licensing and English vocabulary. I will make an INFRA JIRA and ask if an exception can be made for OpenJFX as a dependency (vs. bundled with the JRE, which is allowed).
82. [~steve_rowe] Thank you! Hope there will be good news.
83. I created https://issues.apache.org/jira/browse/LEGAL-396 to ask for an exception to allow Lucene to depend on OpenJFX.
84. Hi [~steve_rowe] thanks for your support with filing the ticket. Looking to solve this one way or another. Thanks [~Tomoko Uchida] for your contribution and research so far!
85. **body:** Hi, I found a similar issue in ASF LEGAL. "Should GPL+CPE move to Cat B?" [https://issues.apache.org/jira/projects/LEGAL/issues/LEGAL-336] Seems the answer is definitely NO. When sticking to JavaFX we probably cannot donate Luke to Apache products, though I like sophisticated JavaFX API and beautiful components. I'm now planning to port the UI layer from JavaFX to Swing. Luke is getting more and more popular in GitHub users (dear committers, how many of you use Luke now?) I'm not sure why a desktop application gains popularity in the web browser era. But if many users need Luke, I think it should become an official member of Lucene in any form...
    label: code-design
86. Random, out of left field question here, please feel free to ignore, but have you considered porting to an HTTP/HTML gui? It could still be a standalone app using a little embedded HTTP service like [this one] (https://docs.oracle.com/javase/7/docs/jre/api/net/httpserver/spec/com/sun/net/httpserver/HttpServer.html). I for one have wanted to run Luke on a remote "cloud" server, but using X Windows to bring up the GUI locally is a pretty big barrier (especially if your local machine is not UNIX!).

87. [~sokolov] Thanks comments, sure, a tiny http server is always on my mind. (I am a just another web engineer, I know there are plenty of nice & light webapp frameworks...) However, I'm not sure that adding web assets (html, css, js) to Lucene, pure Java library, is acceptable. May be server side rendering (JavaScript free) is ok? I have no idea. But web GUI perhaps get the point, when we are in a little bit dark period for client applications now. Apache Pivot, first Mark Miller introduced here, has not been updated for one year. So I've switched to JavaFX as a promising framework, this is decoupled from JDK and blocked for Apache projects (at least for now.)
88. Sorry for unfocused comment, I have no idea about what's the best. So Swing based UI (plus web one, if someone can work for this) could be the starting point because it is not likely to obsoleted or removed from JDK in a while :)
89. Couple of points to consider: * JDK 11 is a disruption moment anyway, as Oracle will no longer provide the official JDK to non-paying customers. Instead, there will be (Redhat-like) policy of OpenJDK vs Official JDK: [http://www.oracle.com/technetwork/java/javase/eol-135779.html] So, we would have to have some real thinking/testing around what that means. * If Luke is supposed to be part of Solr distribution, we already have HTML/CSS assets for Admin UI, etc. Luke could potentially be a plugin of sorts (which has its own issues too) * If Luke is supposed to be part of Lucene-only distribution, I guess the discussion is a bit more complicated
90. Please don't make it a part of solr. The benefit of Luke is to have it standalone so you can use it just with plain Lucene. I am not using solr, but Lucene, so I would like to see it only with Lucene distribution and not be dependent on solr. My 2 cents about this.
91. Sure, this is LUCENE's issue ;)
92. [~arafalov] thanks for your input! Can you please elaborate on 'If Luke is supposed to be part of Lucene-only distribution, I guess the discussion is a bit more complicated' ?
93. I just meant that Lucene does not have a server+HTML+CSS, so any UI addition will be a non-trivial discussion. I did not have any deeper insight than that.
94. Well..., I am now working for completing a java desktop Luke, in any case. If someone is strongly interested in a http server & web app version, feel free to fork the github repository (https://github.com/DmitryKey/luke) and develop it. I am the one long for a web version Luke that can access indexes in remote server, and will glad to cooperate.
95. [~sokolov] hi, I found a web based Luke project, with a rest api server and modern js technologies. This might be what you are wanting? [http://www.flax.co.uk/blog/2017/02/24/release-1-0-marple-lucene-index-detective/] I've not tried this yet, for your information.
96. Hi, as we announced in the Lucene/Solr mailing lists, Luke was re-implemented on top of Swing. [https://github.com/DmitryKey/luke|https://github.com/DmitryKey/luke] The code is licensed under ALv2 and Swing is the part of JDK, so I think there is no obstacle to make it a Lucene submodule. I would like to create another patch and restart this issue, after just fixing styles and colors. The draft patch will be ready for review in the next few weeks or so but I am not sure about when I should cut the feature branch for it. (Seems like Lucene 8.0 release workflow will be kicked off soon.) Don't I have to care about the major version release procedure? Can you please give me some advice? Thanks.
97. bq. as we announced in the Lucene/Solr mailing lists, Luke was re-implemented on top of Swing. Wow, that was fast, congratulations! bq. The code is licensed under ALv2 and Swing is the part of JDK, so I think there is no obstacle to make it a Lucene submodule. +1 bq. The draft patch will be ready for review in the next few weeks or so but I am not sure about when I should cut the feature branch for it. (Seems like Lucene 8.0 release workflow will be kicked off soon.) Don't I have to care about the major version release procedure? Once you put up a patch here I can commit it to the branch I already cut for this issue: {{jira/lucene-2562}} (mentioned in a previous comment of mine on this issue). Depending on when everything's wrapped up, it may or may not make the 8.0 release; if not, then it will be part of whichever release is upcoming. So no, I don't think you need to worry about release procedures for this issue. Is there something specific you're concerned about?
98. Hi Steve, thank you for the quick response. {quote}Is there something specific you're concerned about? {quote} No, I am just wondering that there is something I have to consider. I will create a pull request for review after the next release, Luke 7.6.0. (or 8.0.0?) :)
99. bq. I will create a pull request for review after the next release, Luke 7.6.0. (or 8.0.0?) Please don't wait for a release; Lucene/Solr development happens independently from release cycles.
100. bq. Please don't wait for a release; Lucene/Solr development happens independently from release cycles. I understand, thanks!
101. I created another pull request for review, could you take a look this. -[https://github.com/apache/lucene-solr/pull/490]- [https://github.com/apache/lucene-solr/pull/512] I fixed the code to pass ant validation tasks, following your comments. (This time, Luke module passes most of the validations, validate-source-patterns, forbiddenapis, check-lib-versions and so on. Also unit tests seem to be more stable than before.) {quote}1. validate-source-patterns complains about non-static-final loggers: {quote} The code of the PR has passed {{validate-source-patterns}} on my PC (MacBook / Java 1.8.0_181). {quote}2. It should be possible to run Luke from a source checkout, but it's not now. {quote} {quote}3. I think Luke should be packaged with all other artifacts in Lucene's binary packages, but currently ant package-tgz and -zip don't include everything {quote} {{luke/build.xml}} in this PR creates a simple directory structure as follows in the tgz/zip packeges. {code:java} $ cd lucene-8.0.0-SNAPSHOT $ tree -L 1 luke/ luke/ â”œâ”€â”€ lib â”œâ”€â”€ lucene-luke-8.0.0-SNAPSHOT.jar â”œâ”€â”€ luke.bat // this does not exist yet (TBD). â””â”€â”€ luke.sh {code} {{luke.sh}} can be launched from anywhere. The script knows classpaths (relative paths to jars it depends on) to run. I am not sure that this launch script is the best design choice... if there are better ways to package Luke with Lucene, please let me know. {code:java} # if you are in `lucene-8.0.0` directory: $ ./luke/luke.sh {code} {quote}4. forbidden-apis is very unhappy about classes in javafx.* packages {quote} Besides javafx, a bunch of fixes were needed to pass the check but now it has passed forbiddedapis check. ---- I have some issues to ask or consult with you. 1. Luke is a GUI tool so I think Javadoc should not be published with the tar/zip. I have tried to disable {{javadoc}} and {{documentation-lint}} task, but adding this lines is not working. Am I missing something? {{documentation-lint}} is failed so it does not pass {{ant precommit}}. :/ {code:java} <target name="javadocs"/> <target name="documentation-lint"/> {code} 2. I have noticed that Guice (cglib) has a compatibility issue with JDK9+. Please see [https://github.com/DmitryKey/luke/issues/125] With the latest version of Guice (4.2.2), JDK 11 still emits the warnings. We will be able to remove the dependency on Guice, but if possible I'd like to use the DI framework to keep UI components and model classes loosely coupled. (This greatly helped me when switching JavaFX to Swing, though I didn't expect that situation.)
102. {quote}2. It should be possible to run Luke from a source checkout, but it's not now. {quote} Oh, maybe I missed your intent, after compiling Luke still cannot be quickly started from the terminal. I will deal with this (I think some ant target that launches Luke will be needed for it.) JFYI: For now, Luke can be launched from IDE's Run/Debug feature. I use it when developing Luke. !ã,¹ã,¯ãƒªãƒ¼ãƒ³ã,·ãƒ§ãƒƒãƒˆ 2018-11-05 9.19.47.png|width=100%!
103. {quote}I have noticed that Guice (cglib) has a compatibility issue with JDK9+. {quote} Answering to myself: I looked through my code again and found that (after dozens of refactoring,) current code of Luke does not so heavily depend on Guice. It's not too difficult to completely remove the dependency on Guice, so let me know if you don't like it for the project. :)
104. A small update: > 2. It should be possible to run Luke from a source checkout, but it's not now. I added an ant tartget, {{launch}}, to the build file. This command launches luke after a source checkout. {code:bash} ant -f lucene/luke/build.xml launch {code}
105. Some improvements in the pull request: 1. The luke module passed ant precommit. I examined the Javadoc linter scripts and found that excluding a specific module when executing {{documentation-lint}} is not allowed for now. I added missing javadocs to pass linter's check so that {{ant precommit}} is successfully built. 2. JVM warnings when running on JDK9+ was resolved. The warnings ("An illegal reflective access operation") was not a compatibility issue of Guice. Guice works fine with JDK9+. I added {{--add-opens}} JVM option in the start script to allow deep reflection. By doing so, dependency injection works within the module system and the warnings disappears. 3. A start script for windows was added. {{luke.bat}} was added for windows os. Is there anything else I can do for this issue? ;D
106. Hi, I updated the pull request to put in the latest fixes on Luke: https://github.com/apache/lucene-solr/pull/512 Could someone (may be [~steve_rowe]?) please review the pull request. Of course I'm not in a hurry, just hope this will be merged in the near future. Today I found Luke project on github earns 1.1k+ stars. I am very pleased to see that and also feel like this is enough for us... Thanks.
107. Why do we need to do this? bq. The warnings ("An illegal reflective access operation") was not a compatibility issue of Guice. Guice works fine with JDK9+. I added --add-opens JVM option in the start script to allow deep reflection. By doing so, dependency injection works within the

module system and the warnings disappears. That's likely to break in later Java versions, so do not depend on deep reflection on core classes! Forbiddenapis should not allow this, so where does it happen? Inside Guice? Uwe

108. bq. Forbiddenapis should not allow this, so where does it happen? Inside Guice? Without {{--add-opens}} option, JDK complains as follows when launching Luke: {code:java} WARNING: An illegal reflective access operation has occurred WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/tmp/lucene-8.0.0-SNAPSHOT/luke/lib/guice-4.1.0.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java.security.ProtectionDomain) WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1 WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations WARNING: All illegal access operations will be denied in a future release {code} It occurs inside cglib and there is an issue about this. [https://github.com/google/guice/issues/1133] Discussion is open but seems there is no clear answer from Guice developers. Should we remove the dependency to Guice / cglib?
109. I will remove the dependency to Guice so that the {{--add-opens}} option can be removed. (I might be wrong..., but in my understanding Luke can safely use the dependency injection framework without this special option if we modularize Lucene in the Jigsaw's way and open only luke module to Guice, but it should not be a viable option for now.)   If there are any other obstacles to merge the patch please point them out to me. I will fix it following your review.
110. **body:** Hi Tomoko, I would also favour to remove Guice, if it's easy to do. By the way, I opened the CGLIB issue back at that time: https://github.com/cglib/cglib/issues/93 Some background: In early Java 9 version deep reflection was not working at all, which was relaxed before release. The "official" statement is: CGLIB is no longer maintained, people should migrate to ByteBuddy. Because of this we removed all our Mocking frameworks (we had many in Solr) and moved everything to Mockito. Mockito uses ByteBuddy and uses a more modern way to produce proxies. I really don't like to make Lucene depend on a library (CGLIB) that's no longer maintained by the community. If we want to use Guice, Guice should move to ByteBuddy. This is one reason why Elasticsearch dropped dependency injection with Guice long ago. That's also too much magic and should be avoided. +1 for the patch without Guice. Once you are done, I will look over it again. I mentioned a small thing on the Pull Request: Please don't allow writes to sysprops GLOBALLY! There is a reason why tests are not allowed to do this. If You really need to set a system property please document it and list those exemptions separately in the policy file (with comment).
    label: code-design
111. **body:** Thank you, Uwe. {quote}I would also favour to remove Guice, if it's easy to do. {quote} Yes, it should be technically easy to remove Guice though some work is needed. Also I'd better to remove a few more libraries (ini4j and FindBugs) which are no longer maintained. {quote}I mentioned a small thing on the Pull Request: Please don't allow writes to sysprops GLOBALLY! {quote} (If I remember right) I modified the policy file because unit tests encountered errors come from log4j. I will try to find workarounds for it, so let me discuss with you about the details later.
    label: code-design
112. Hi, I have created another pull request, this is a fork from the latest version of Luke (8.0.0). [https://github.com/apache/lucene-solr/pull/618] The changes from the previous one: - Removed the dependency on Guice and hence {{--add-opens}} option was gone. (It has been replaced with ordinary static factory methods.) - Removed the dependency on FindBugs. (Instead of @Nonnull annotation runtime null checks were added.) - Removed the dependency on ini4j. (Instead I've written a tiny config file writer/reader.) [~thetaphi] Can you please review that again? And I added these lines to the security manager policy file to run test cases for some classes that include log4j logger, although I didn't need to allow any writes to system properties. I have not yet checked the details but just followed the failure messages from JUnit. {code:java} // needed by org.apache.logging.log4j permission java.lang.RuntimePermission "getenv.*"; permission java.lang.RuntimePermission "getClassLoader"; permission java.lang.RuntimePermission "setContextClassLoader"; permission javax.management.MBeanServerPermission "createMBeanServer"; {code}
113. I would like to remove one more library {{reflections.}} I've found it still uses relatively old versions of other 3rd party libraries, that could cause problems in the future - you might already be aware of this. By doing so, Luke will depend only on Lucene itself and logger libraries (log4j and slf4j). ---- By the way, to be honest, (while I've worked for two years or so for this issue) currently I am not sure whether integrating Luke into Lucene is welcomed by the community. So please let me know if it's preferable to keep Luke separated from Apache Lucene project. I don't intend to overturn our achievement so far, Luke's source code has been well modernized and slimmed down through feedbacks here. My attempts were worth just because that. Of course I hope my worries are groundless (maybe because I have had no chance to ask/talk with committers about the issue in person) and would appreciate your feedbacks as usual.
114. Hi, bq. By the way, to be honest, (while I've worked for two years or so for this issue) currently I am not sure whether integrating Luke into Lucene is welcomed by the community. I'd really like to get this into the Lucene source, code so it can be released together with Lucene and servers vor better out-of-box experience. I will check yur latest patch again. Sorry for being not so responsive, I was on travel yesterday! bq. I would like to remove one more library reflections. I've found it still uses relatively old versions of other 3rd party libraries, that could cause problems in the future - you might already be aware of this. By doing so, Luke will depend only on Lucene itself and logger libraries (log4j and slf4j). This is great!
115. Thank you, and sorry I was a little bit overwhelmed by this long running issue. I will update the patch to remove the dependency on {{reflections.}}
116. No problem, I took the issue and will take care of pushing that change.
117. Hi, I updated the patch so that the dependency on {{reflections}} is removed. Instead, its own jar files scanner {{o.a.l.luke.util.reflection.ClassScanner}} has been added to make Luke work as before. Also transitive dependencies (Guava etc.) were removed. Please check them when you have time.
118. Hi [~Tomoko Uchida], thanks for adding the patch. I am now a bit confused about the pull requests related to that patch. Which one is the uptodate one? I made a comment on one of the pull requests (https://github.com/apache/lucene-solr/pull/618), but not sure if this is reflected in the patch. I'd like to remove the complexits in isSubclassOf, as this can be handled easily with standard java code.
119. Nevertheless, we should clean up the attached files, there is no patch! I only see the pull request: https://github.com/apache/lucene-solr/pull/618
120. Thanks [~thetaphi], {quote}Nevertheless, we should clean up the attached files, there is no patch! I only see the pull request: [https://github.com/apache/lucene-solr/pull/618] {quote} Yes, this is the latest pull request, sorry for the confusion. I am not sure how to clean-up the old attachments. Should I create a patch including all the changes and attach to the issue?
121. **body:** The confusion came from the fact that you said "I updated the patch..." and I was looking for a patch. You did not add a link to the PR, I had to manually find it in my mails. We should only upload a patch at the end, for now you just have to know that the attached patches are all no longer valid. Would it be possible to remove the isSubclass() code and replace by the JDK one? The cache is also obsolete.
    label: code-design
122. {quote}Would it be possible to remove the isSubclass() code and replace by the JDK one? {quote} I think you mean {{Class.isAssignableFrom()}}, I have not noticed this method. I will remove isSubclass() by replacing that. {quote}I made a comment on one of the pull requests ([https://github.com/apache/lucene-solr/pull/618]), but not sure if this is reflected in the patch. {quote} I cannot find your comments in the past PRs, 'cause there are lots of files and GitHub search {{commenter:uschindler}} does not work for finding them... :'(Can you please tell me the points again?
123. **body:** I screenshotted it: !screenshot-1.png! Basically, I'd remove the whole ConcurrentCache and all that stuff, as checking if a class is a subtype of another one is very easy with {{Class#isAssignableFrom(Class)}}.
    label: code-design
124. I updated the pull request ([https://github.com/apache/lucene-solr/pull/618]) as you suggested: isSubclass() and ConcurrentHashMap property were replaced with {{Class#isAssignableFrom.}} There is no other changes. ([diff|https://github.com/apache/lucene-solr/pull/618/commits/8edfa290328c5a6a29450ac1eefc7a68a699e35d]) Also I added a link to this issue from the PR, so that an issue link appears in "links to" section. [~thetaphi] Are there any other things that we should do, apart from IDE setups for Eclipse/NetBeans? (I'd like to add them when the review is done.)
125. Hi [~Tomoko Uchida], I have noticed the change already! It looks fine. I will now do some tests, but according to my knowledge this looks fine. It's an antipattern to load every class on classpath and collect those that are a specific subclass, but unfortunately there is no way in Java to do this solely

through classloaders (they don't allow to list classes). The code should work fine with Java 8 and 11 and later, as long as Luke is shipped as a JAR file. Once we switch to Java module system this may change, but it's OK for now. I like the code to find the JAR files that have a package by doing a ClassLoader#getResource() and analyzing the URL. I will do some tests on Lucene master (9.0) and if all looks fine, I can merge it soon. Do you have any other plans except adding the Netbeans/Exclipse files? I think Eclipse should work out of box (I will check this, trust me), but I have no idea about Netbeans&Co. Uwe

126. One small thing: why do you compare the "class names" near isAssignableFrom(). I think it's enough to use Objects.equals(class1, class2) - because 2 classes with same name can still be different (not in this case, but in general, e.g. for compiled expressions in Lucene). I can't really review the GUI code as I have no idea about Swing and AWT, I just trust you that it confirms to standards. :-)
127. **body:** Another small thing: in the log4j config, you can remove the reflections config (ERROR). And I think in production we should only use INFO logging?
    label: code-design
128. **body:** On a first test with Windows: all looks fine. There are only some glitches with alignment of buttons, but this may be caused by 150% HiDPI. I can confirm, that it's better to use Java 11, as Java 8 does not zoom according to system settings. Great work. I am now clicking through the folders and check for licenses and files. Do we have a license for the TTF (font) file? I think we may have to add it at least to the NOTICE file. Opinions? Some thing that's a bit unrelated: When you open an index, you can select the directory implementation, but it also offers WindowsDirectory and NativeUnixDirectory, although they don't work, as the .so/.dll files are missing (we don't even compile them). IMHO we should think how to proceed with those - it's unrelated, the class discovery just found it.
    label: code-design
129. **body:** I was also able to compile and start with "ant launch" from a completely cleaned build folder. So IVY/ANT module dependencies work.
    label: code-design
130. **body:** The start files in bin are intended to only work in the binary release ZIP, right? I have another suggestion, but we can delay that to later: We should not ship the resources directories "img" and "font" and the messages files in the root folder of the JAR file and instead move them to "org.apache.lucene.luke.(img|font|messages)", also load them using a relative path instead of a full path in the code (use Class.getResource with a relative path). And I'd also suggest to not put the log4j2.xml file with its default name into the JAR file. If some user adds all jar files from the Lucene ZIP file to his classpath (some users do this, I see it every day), he will import that log4j config file and break his project. As the logging is handled inside the app, I'd suggest to move the config file also to Luke's package and load it explicitely with Log4J bootstrapping in the main() method. Otherwise I am happy now :-)
    label: code-design
131. When running tests, I get: {noformat} [junit4] > Caused by: java.security.AccessControlException: access denied ("java.io.FilePermission" "C:\Users\Uwe%20Schindler\Projects\lucene\trunk-lusolr1\lucene\build\luke\classes\test\log4j2-test.xml" "read") [junit4] > at java.security.AccessControlContext.checkPermission(AccessControlContext.java:472) [junit4] > at java.security.AccessController.checkPermission(AccessController.java:884) [junit4] > at java.lang.SecurityManager.checkPermission(SecurityManager.java:549) [junit4] > at java.lang.SecurityManager.checkRead(SecurityManager.java:888) [junit4] > at java.io.File.exists(File.java:814) [junit4] > at org.apache.logging.log4j.core.util.FileUtils.fileFromUri(FileUtils.java:88) [junit4] > at org.apache.logging.log4j.core.util.FileUtils.fileFromUri(FileUtils.java:88) [junit4] > at org.apache.logging.log4j.core.config.ConfigurationSource.fromResource(ConfigurationSource.java:281) [junit4] > at org.apache.logging.log4j.core.config.ConfigurationFactory$Factory.getConfiguration(ConfigurationFactory.java:449) [junit4] > at org.apache.logging.log4j.core.config.ConfigurationFactory$Factory.getConfiguration(ConfigurationFactory.java:382) [junit4] > at org.apache.logging.log4j.core.config.ConfigurationFactory.getConfiguration(ConfigurationFactory.java:261) [junit4] > at org.apache.logging.log4j.core.LoggerContext.reconfigure(LoggerContext.java:616) [junit4] > at org.apache.logging.log4j.core.LoggerContext.reconfigure(LoggerContext.java:637) [junit4] > at org.apache.logging.log4j.core.LoggerContext.start(LoggerContext.java:231) [junit4] > at org.apache.logging.log4j.core.impl.Log4jContextFactory.getContext(Log4jContextFactory.java:153) [junit4] > at org.apache.logging.log4j.core.impl.Log4jContextFactory.getContext(Log4jContextFactory.java:45) [junit4] > at org.apache.logging.log4j.LogManager.getContext(LogManager.java:155) [junit4] > at org.apache.logging.slf4j.Log4jLoggerFactory.getContext(Log4jLoggerFactory.java:43) [junit4] > at org.apache.logging.log4j.spi.AbstractLoggerAdapter.getLogger(AbstractLoggerAdapter.java:46) [junit4] > at org.apache.logging.slf4j.Log4jLoggerFactory.getLogger(Log4jLoggerFactory.java:29) [junit4] > at org.slf4j.LoggerFactory.getLogger(LoggerFactory.java:358) [junit4] > at org.slf4j.LoggerFactory.getLogger(LoggerFactory.java:383) [junit4] > at org.apache.lucene.luke.models.commits.CommitsImpl.<clinit>(CommitsImpl.java:44) [junit4] > ... 37 more {noformat}
132. I found the license for the font, all fine! There are some obsolete license files that have to be removed: guava, javassist
133. **body:** About the security exception: The whole issue comes from the fact that loig4j tries to load the file instead of using the ClassLoader, it reads it directly using "File" class. I don't like to add read access to everywhere in Lucene's policy file! How about the following suggestion - also related to my issue with shipping a log4j2.xml in the root folder of the JAR file and the more or less static config: Can we not remove the log4j config files completely and instead hardcode the "textarea" appender in the startup code of the application. There is no reason to have a config file, just configure log4j as part of startup. For tests I'd do the same, just configure Log4J to don't log at all in the initializer of the test base class (just create a superclass for all your tests).
    label: code-design
134. Thanks, I made this changes as suggested. * Use Objects#equals() instead of String comparison * Fixes log4j.xml {quote}There are only some glitches with alignment of buttons, but this may be caused by 150% HiDPI. I can confirm, that it's better to use Java 11, as Java 8 does not zoom according to system settings. {quote} Yes, Java 9+ is recommended for HiDPI displays on Windows/Linux ([http://openjdk.java.net/jeps/263]). I think we need some notes for this. {quote}Do we have a license for the TTF (font) file? I think we may have to add it at least to the NOTICE file. Opinions? {quote} We already have License and NOTICE file for ([The Elegant Icon Font |https://www.elegantthemes.com/blog/resources/elegant-icon-font]): {{licenses/elegant-icon-font-LICENSE-MIT.txt}} and {{licenses/elegant-icon-font-NOTICE.txt}} {quote}Some thing that's a bit unrelated: When you open an index, you can select the directory implementation, but it also offers WindowsDirectory and NativeUnixDirectory, although they don't work, as the .so/.dll files are missing (we don't even compile them). {quote} I missed it and I'm not familiar with those directories. I think we can exclude those two directories from the ComboBox for now (because ordinary users wouldn't need them,) or should we add notes for the native libraries (or detailed error messages when they are missing)?
135. {quote}There are some obsolete license files that have to be removed: guava, javassist {quote} I have removed those obsolete licenses when I removed \{{reflections}}, just for the record :)
136. The SecuirtyException is a bug in Log4J: I have whitespace in my path name and when loading the file it incorrectly gets the url-encoded path. It would fail anyways later (with a FileNotFoundError)! The bug is in {{org.apache.logging.log4j.core.util.FileUtils.fileFromUri}}: https://logging.apache.org/log4j/2.0/log4j-core/apidocs/src-html/org/apache/logging/log4j/core/util/FileUtils.html#line.62 That's brokenness as mentioned in the comment above the method! God damn... Keep in mind that our Jenkins servers are partially also using whitespace in path names exactly to test those issues.
137. **body:** bq. I have removed those obsolete licenses when I removed reflections, just for the record OK, they are gone now. Seems like it was a GIT problem. bq. I missed it and I'm not familiar with those directories. I think we can exclude those two directories from the ComboBox for now (because ordinary users wouldn't need them,) or should we add notes for the native libraries (or detailed error messages when they are missing)? I don't think we should add hacks like this to the GUI. Let's just ignore this issue for now. IMHO, we should find a better solutions, because it's stupid

to ship a compiled class file with Lucene that does not work at all. We should remove it in a separate issue or just provide it as uncompiled example code.
label: code-design

138. {quote}The start files in bin are intended to only work in the binary release ZIP, right? {quote} Yes, I think it's OK because developers can launch Luke by {{ant launch}} command. Do you have any concerns about this? bq. I have another suggestion, but we can delay that to later: We should not ship the resources directories "img" and "font" and the messages files in the root folder of the JAR file and instead move them to "org.apache.lucene.luke.(img|font|messages)", also load them using a relative path instead of a full path in the code (use Class.getResource with a relative path). bq. And I'd also suggest to not put the log4j2.xml file with its default name into the JAR file. If some user adds all jar files from the Lucene ZIP file to his classpath (some users do this, I see it every day), he will import that log4j config file and break his project. As the logging is handled inside the app, I'd suggest to move the config file also to Luke's package and load it explicitely with Log4J bootstrapping in the main() method. bq. How about the following suggestion - also related to my issue with shipping a log4j2.xml in the root folder of the JAR file and the more or less static config: Can we not remove the log4j config files completely and instead hardcode the "textarea" appender in the startup code of the application. There is no reason to have a config file, just configure log4j as part of startup. Thanks, I will update the PR as suggested.
139. bq. Yes, I think it's OK because developers can launch Luke by ant launch command. Do you have any concerns about this? +1
140. Hi [~Tomoko Uchida], I have some smaller changes while going through the whole code, but its relatively hard to all describe them in the issue or here. Would it be possible to get write access to your Github repositoy, so we can both work on finalizing it? Alternatively I can push your branch to Apache's GIT, but you have no push access there, so I'd propose to finalize this on Github and then merge it to Lucene "master". It's just some smaller changes (wrong use of a charset, an obsolete log4j-1.2-api dependency, and maybe some smaller changes I'd like to do). I can also help with fixing the log4j issues.
141. Just a status update: I finished reviewing and improving some code in the package (on [~Tomoko Uchida]'s branch). I am very happy with the current state, it works fine, all functionality seem to work. It builds fine. Eclipse also opens it correctly (did not check the other IDEs). It passes precommit, license files seem uptodate. We minimzed dependencies to only log4j2, configured inside the application and logging to a panel in the app. During tests, log4j2 runs without any config, so it suppresses logs. I minimzed changes to Lucene's testing policy file, but IMHO, I will open a separate issue to allow for Lucene/Solr modules to add some extra "opens" not applied to the basic policy file. Solr currently has its own pone, but this can be unified now. I also tested the GUI with Windows, some UI glitches are still there (150% scale). We will wait over the weekend and then merge the branch into Lucene master. I think it should also go into 8.1 - any comments?
142. **body:** I just want to add a supplement comment: IDE setup for IntelliJ IDEA "{{ant idea}}" is fine, I usually use IDEA when developing Luke. I have not confirmed those for NetBeans yet.
    label: code-design
143. Here is the final patch including all commits on [~Tomoko Uchida]'s branch: [^LUCENE-2562.patch]
144. I pushed the branch to ASF Github: https://gitbox.apache.org/repos/asf?p=lucene-solr.git;a=shortlog;h=refs/heads/jira/lucene-2562-luke-swing-3 If you have more changes i will merge them in.
145. Hi [~Tomoko Uchida], as you are a committer now on Apache Lucene, I assigned the issue to you. I am still here to help when merging it to our branches, adding CHANGES.txt,... Uwe
146. Hi [~Tomoko Uchida], I prepared the branch to be squash-merged. Once you have your ASF account, I would propose that you do the last step on your own. Here is the steps to do it: - Checkout the branch from ASF gitbox (or use your own one from Github), does not matter. Be sure to pull latest updates!!! - Make sure to configure the GIT checkout to use your {{@apache) mail address for commits (it's also wise to add it as an alias to github, so Github and ASF bots can identify you as the same person). It may be wise to make the mail address the default for all Lucene checkouts you maintain. Also it's wise to have "origin" set to ASF gitbox and Github as alternative remote (e.g., "mocobeta") - Switch back to master. Be sure to pull latest updatesfrom ASF gitbox!!! - Run: {{git merge --squash jira/lucene-2562-luke-swing-3}} - Commit the stuff with a useful commit message: {{git commit -m 'LUCENE-2562: Add Luke as a Lucene module'}} - Push changes - Switch to branch "branch_8x" (the 8.x development branch) - Cherry-pick the previous commit from master branch, or alternatively do the same squashing merge. - Push changes Good luck - I am here for help. Uwe
147. I also gave you more rights in Apache JIRA and aded you as the assignee for this issue.
148. Commit 3a78c1999a8716669c574448b2b07c0f63cbb39c in lucene-solr's branch refs/heads/master from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=3a78c19] LUCENE-2562: Add Luke as a Lucene module
149. Commit 9915056cbc8380658ae2cf4e413969642d96a815 in lucene-solr's branch refs/heads/branch_8x from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=9915056] LUCENE-2562: Add Luke as a Lucene module
150. Hi [~Tomoko Uchida], Great work! Thanks for pushing the changes as your first action as committer. It was a very impressive change! I hope Jenkins is happy, if not we can fix later. Uwe
151. Commit 3a78c1999a8716669c574448b2b07c0f63cbb39c in lucene-solr's branch refs/heads/jira/LUCENE-8738 from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=3a78c19] LUCENE-2562: Add Luke as a Lucene module
152. Hi [~Tomoko Uchida], The Luke module has Maven-related build problems. E.g. from https://builds.apache.org/job/Lucene-Solr-SmokeRelease-master/1306/ : {noformat} BUILD FAILED /x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/build.xml:446: The following error occurred while executing this line: /x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/lucene/build.xml:433: The following error occurred while executing this line: /x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/lucene/build.xml:413: The following error occurred while executing this line: /x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/lucene/common-build.xml:2269: The following error occurred while executing this line: /x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/lucene/common-build.xml:1727: The following error occurred while executing this line: /x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/lucene/common-build.xml:657: Unable to initialize POM pom.xml: Could not find the model file '/x1/jenkins/jenkins-slave/workspace/Lucene-Solr-SmokeRelease-master/lucene/build/poms/lucene/luke/pom.xml'. for project unknown {noformat} This is caused by the Luke module not being set up to produce Maven artifacts (e.g. there is no POM template under {{dev-tools/maven/}}). I don't think publishing binary/source/javadoc jars on Maven central would be useful, but the build should be fixed. The following addition to {{lucene/luke/build.xml}} fixes the problematic targets ({{validate-maven-dependencies}} and {{generate-maven-artifacts}}) by locally re-defining Maven-related targets to do nothing: {noformat} <!-- Luke has no Maven artifacts --> <target name="-dist-maven"/> <target name="-install-to-maven-local-repo"/> <target name="-validate-maven-dependencies"/> <target name="-append-module-dependencies-properties"/> {noformat}
153. Hi [~steve_rowe], I've not noticed the Maven setups. I will fix it soon as suggested. A novice question: Can I add another patch (or pull request) to this issue, or should I create another issue?
154. Ah, there is still the branch for this issue: [https://github.com/apache/lucene-solr/tree/jira/lucene-2562-luke-swing-3] I will update this branch and merge it to master & branch_8x.
155. Yeah that's fine. Just merge it or cherry pick, whatever you think is better.
156. [~steve_rowe]: would the smoke tester complain if we have a module without maven artifacts? I am not sure about that, mabye we just wait for the next run of nightly-smoke. If it works all fine, if not we have to think. Nevertheless, [~Tomoko Uchida]: did you publish maven artfacts for the original "luke"? If not we are fine to disable it, but if the original Luke was published on Maven, we should maybe do the same here, too.
157. I pushed the change of {{lucene/luke/build.xml}} to the branch: https://github.com/apache/lucene-solr/commits/jira/lucene-2562-luke-swing-3 Ant target "generate-maven-artifacts" was built successfully but "validate-maven-dependencies" was failed on my PC, with the error that seems to have nothing to do with luke ... {code:bash} -validate-maven-dependencies: [artifact:dependencies] An error has occurred while processing the Maven artifact tasks. [artifact:dependencies] Diagnosis: [artifact:dependencies] [artifact:dependencies] Unable to resolve artifact: Unable to get dependency information: Unable to read the metadata file for artifact 'com.healthmarketscience.jackcess:jackcess:jar': Cannot find parent: com.healthmarketscience:openhms-parent for project: com.healthmarketscience.jackcess:jackcess:jar:2.1.12 for project com.healthmarketscience.jackcess:jackcess:jar:2.1.12 [artifact:dependencies] com.healthmarketscience.jackcess:jackcess:jar:2.1.12

[artifact:dependencies] [artifact:dependencies] from the specified remote repositories: [artifact:dependencies] maven-restlet (http://maven.restlet.org), [artifact:dependencies] central (http://repo1.maven.org/maven2), [artifact:dependencies] releases.cloudera.com (https://repository.cloudera.com/artifactory/libs-release), [artifact:dependencies] apache.snapshots (foobar://disabled/) [artifact:dependencies] [artifact:dependencies] Path to dependency: [artifact:dependencies] 1) org.apache.solr:solr-dataimporthandler-extras:jar:9.0.0-SNAPSHOT [artifact:dependencies] [artifact:dependencies] [artifact:dependencies] Not a v4.0.0 POM. for project com.healthmarketscience:openhms-parent at /home/moco/.m2/repository/com/healthmarketscience/openhms-parent/1.1.8/openhms-parent-1.1.8.pom BUILD FAILED /home/moco/repo/lucene-solr/build.xml:235: The following error occurred while executing this line: /home/moco/repo/lucene-solr/solr/build.xml:735: The following error occurred while executing this line: /home/moco/repo/lucene-solr/solr/common-build.xml:479: The following error occurred while executing this line: /home/moco/repo/lucene-solr/solr/common-build.xml:382: The following error occurred while executing this line: /home/moco/repo/lucene-solr/lucene/common-build.xml:692: Unable to resolve artifact: Unable to get dependency information: Unable to read the metadata file for artifact 'com.healthmarketscience.jackcess:jackcess:jar': Cannot find parent: com.healthmarketscience:openhms-parent for project: com.healthmarketscience.jackcess:jackcess:jar:2.1.12 for project com.healthmarketscience.jackcess:jackcess:jar:2.1.12 com.healthmarketscience.jackcess:jackcess:jar:2.1.12 from the specified remote repositories: maven-restlet (http://maven.restlet.org), central (http://repo1.maven.org/maven2), releases.cloudera.com (https://repository.cloudera.com/artifactory/libs-release), apache.snapshots (foobar://disabled/) Path to dependency: 1) org.apache.solr:solr-dataimporthandler-extras:jar:9.0.0-SNAPSHOT {code} ---- bq. Nevertheless, Tomoko Uchida: did you publish maven artfacts for the original "luke"? If not we are fine to disable it, but if the original Luke was published on Maven, we should maybe do the same here, too. No, we have not published any maven artifacts of luke. I agree with [~steve_rowe].

158. [~thetaphi]: I didn't think of running the smoke tester; I'm not sure whether/how Maven artifact validation checks artifact correspondence, I'll run it locally on the branch now that Tomoko has updated it. [~Tomoko Uchida]: I saw that same problem, as did Kevin Risden [over on SOLR-9515|https://jira.apache.org/jira/browse/SOLR-9515?focusedCommentId=16762835&page=com.atlassian.jira.plugin.system.issuetabpanels:comment-tabpanel#comment-16762835]. AFAICT this happens when you have a local maven repo that you haven't used the Lucene/Solr Maven build with, but which is populated by the Lucene/Solr Ant build via the Maven Ant Tasks plugin. Earlier in the log I see: {noformat} [artifact:dependencies] Downloading: com/healthmarketscience/openhms-parent/1.1.8/openhms-parent-1.1.8.pom from repository taglets at http://maven.geotoolkit.org/ [artifact:dependencies] Transferring 0K from taglets [artifact:dependencies] [WARNING] *** CHECKSUM FAILED - Checksum failed on download: local = 'c53d5de1975ce58462f226d7ed126e02d8f1f58b'; remote = '<html> [artifact:dependencies] <head><title>301' - RETRYING {noformat} So (for me anyway) what happened was that a Maven repository we don't specify (probably specified in the jackcess POM hierarchy) returned an HTML page and a 301 HTTP code (permanently relocated), which is improperly interpreted by Maven Ant Tasks as the appropriate artifact and saved as if it were the jackcess parent POM. I worked around this by deleting the local repo's bad file, then installing the correct POM manually: {noformat} rm ~/.m2/repository/com/healthmarketscience/openhms-parent/1.1.8/openhms-parent-1.1.8.pom curl -O https://repo1.maven.org/maven2/com/healthmarketscience/openhms-parent/1.1.8/openhms-parent-1.1.8.pom mvn install:install-file -Dfile=openhms-parent-1.1.8.pom -DgroupId=com.healthmarketscience -DartifactId=openhms-parent -Dversion=1.1.8 -Dpackaging=pom {noformat} Maybe there's a better way? (E.g. running the Maven build in this context?) The above fixed the problem for me. (Maven Ant Tasks have been EOL'd for a few years, so there will be no fix for this problem in that project.)
159. **body:** I have seen the maven issue, too (a while back). It looks like for some legal reasons they removed some artifacts from Maven central which are already in your repository. The easiest is to delete the whole local repository... but then it downloads everythinga agin. If this takes too long for you, just clean up from the ~/.m2 cache folders.
label: code-design
160. Thanks, Steve, "ant validate-maven-dependencies" is successfully built on my local PC after replacing the POM with the correct one. I can cherry-pick the change to the master and branch_8x when you confirm it.
161. bq. and a 301 HTTP code (permanently relocated), That's a bug in Java, which wont be fixed. The Java HTTP as shipped with JDK behind the URL class does not follow redirects between different HTTP protocols, so it won't move from http to https. Thats a known issue: [https://bugs.openjdk.java.net/browse/JDK-4620571?focusedCommentId=12159233&page=com.atlassian.jira.plugin.system.issuetabpanels%3Acomment-tabpanel#comment-12159233] This makes me really angry as its against HTTP spec, but Oracle/SUN never implemented it for some fake security reasons. In fact HTTP->HTTPS is legit, just the other way round is not. Not sure we need a workaround for that, but that's a separate issue.
162. All fine, we will wait for jenkins to prove. If we need to change smoke tester we can wait till later.
163. The smoke tester failed for a non-Maven-related reason: {noformat} [smoker] Test Lucene... [smoker] verify sha512 digest [smoker] unpack lucene-9.0.0.tgz... [smoker] Traceback (most recent call last): [smoker] File "/home/sarowe/git/lucene-solr/dev-tools/scripts/smokeTestRelease.py", line 1518, in <module> [smoker] main() [smoker] File "/home/sarowe/git/lucene-solr/dev-tools/scripts/smokeTestRelease.py", line 1448, in main [smoker] smokeTest(c.java, c.url, c.revision, c.version, c.tmp_dir, c.is_signed, c.local_keys, ' '.join(c.test_args)) [smoker] File "/home/sarowe/git/lucene-solr/dev-tools/scripts/smokeTestRelease.py", line 1499, in smokeTest [smoker] unpackAndVerify(java, 'lucene', tmpDir, artifact, gitRevision, version, testArgs, baseURL) [smoker] File "/home/sarowe/git/lucene-solr/dev-tools/scripts/smokeTestRelease.py", line 604, in unpackAndVerify [smoker] verifyUnpacked(java, project, artifact, unpackPath, gitRevision, version, testArgs, tmpDir, baseURL) [smoker] File "/home/sarowe/git/lucene-solr/dev-tools/scripts/smokeTestRelease.py", line 679, in verifyUnpacked [smoker] raise RuntimeError('%s: unexpected files/dirs in artifact %s: %s' % (project, artifact, l)) [smoker] RuntimeError: lucene: unexpected files/dirs in artifact lucene-9.0.0.tgz: ['luke'] {noformat} Sorry, I don't have time right now to work on it, but I will have time this weekend if nobody beats me to it. FYI, I expect there will be a Maven-related problem with the smoke tester - here's what I think will be the problem (from {{smokeTestRelease.py}}) - probably just need to introduce an exception list: {noformat} def verifyPOMperBinaryArtifact(artifacts, version): print(' verify that each binary artifact has a deployed POM...') reBinaryJarWar = re.compile(r'%s\.[jw]ar$' % re.escape(version)) for project in ('lucene', 'solr'): for artifact in [a for a in artifacts[project] if reBinaryJarWar.search(a)]: POM = artifact[:-4] + '.pom' if POM not in artifacts[project]: raise RuntimeError('missing: POM for %s' % artifact) {noformat}
164. The fix for smoker is to be done here in dev-tools/scripts/smoketestRelease.py: {code:python} if project == 'lucene': # TODO: clean this up to not be a list of modules that we must maintain extras = ('analysis', 'backward-codecs', 'benchmark', 'classification', 'codecs', 'core', 'demo', 'docs', 'expressions', 'facet', 'grouping', 'highlighter', 'join', 'memory', 'misc', 'queries', 'queryparser', 'replicator', 'sandbox', 'spatial', 'spatial-extras', 'spatial3d', 'suggest', 'test-framework', 'licenses') {code} The maven part should not fail as it just downloads everything from the maven folder and then checks if POMS are available. But as it's not in the maven folder of the release, it should not fail. I'd wait for that to fail or not fail. [~Tomoko Uchida]: can you just fix the above code and commit that, too. You just need to add "luke".
165. {quote}Tomoko Uchida: can you just fix the above code and commit that, too. You just need to add "luke". {quote} I added 'luke' to the line: [https://github.com/apache/lucene-solr/commit/f85819985b5a9c0c10c0e810c1826cd40be735d2] (did not do any confirmation, just fixed the python code...)
166. [~Tomoko Uchida], I'm re-running {{ant nightly-smoke}} from the branch after your latest commit - you should be able to do the same.
167. {{ant nightly-smoke}} succeeded for me: bq. [smoker] SUCCESS! [0:37:23.510122] Uwe is right - {{verifyPOMperBinaryArtifact()}} just makes sure that each binary artifact *in the {{maven/}} directory* has a POM, and since Luke doesn't put anything in there, no problem is detected. +1 to merge/cherry-pick.
168. I'm now running {{ant nightly-smoke}} on my box. I will cherry pick the two changes (in luke/build.xml and smoketestRelease.py) as soon as the smoker finishes. (Seems it takes a long time on my old PC...)
169. Commit f85c08224b47e10f3482f27f3811b44dcae3be59 in lucene-solr's branch refs/heads/master from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=f85c082] LUCENE-2562: Luke has no Maven artifacts
170. Commit 6e28cd60a8247ad1339bea2ae9dfbb912507594b in lucene-solr's branch refs/heads/master from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=6e28cd6] LUCENE-2562: Fix smoker for 'luke' module.

171. Commit 06f7aff8b1a076602d9de26d5ccf6a8128c388fd in lucene-solr's branch refs/heads/branch_8x from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=06f7aff] LUCENE-2562: Luke has no Maven artifacts
172. Commit 811aae60cd79b9e5e931516219de7e7f10363bed in lucene-solr's branch refs/heads/branch_8x from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=811aae6] LUCENE-2562: Fix smoker for 'luke' module.
173. {{ant nightly-smoke}} also succeeded for me: bq. [smoker] SUCCESS! [1:14:24.029762] I cherry picked the changes to master & branch_8x (as the ASF bot say.)
174. **body:** [~Tomoko Uchida]: Thanks for fixing this! Don't be afraid that your first commit caused such issues! That's just normal. Whenever somebody added a new module in the last few years this always caused many followup commits to fix the distribution like maven & co. This is so complex so you can not easily do this right - not even the long-term committers. There are too many release-specific things that need to be taken care, but Jenkins helps us to find the issues (especially as I personally dont want to block my whole system for several hours running Solr test cases and smoke testers).
     label: code-design
175. Commit f85c08224b47e10f3482f27f3811b44dcae3be59 in lucene-solr's branch refs/heads/jira/LUCENE-8738 from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=f85c082] LUCENE-2562: Luke has no Maven artifacts
176. Commit 6e28cd60a8247ad1339bea2ae9dfbb912507594b in lucene-solr's branch refs/heads/jira/LUCENE-8738 from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=6e28cd6] LUCENE-2562: Fix smoker for 'luke' module.
177. **body:** I forgot to mention about required Java version for Luke. Java 11 (also 9, 10 are ok) is highly recommended (actually it's needed) to run Swing apps on HiDPI graphics on Windows/Linux. I have noted that in Luke's release note like this: [https://github.com/DmitryKey/luke/releases/tag/luke-swing-8.0.0] I think we should mention about this limitation in somewhere until minimum Java version requirement is changed to 11, where should I add such notes? Maybe I would add some texts to Luke's About dialog (and a startup warning popup message would be much better but I have not yet examined if the application itself can detect "HiDPI graphics").
     label: documentation
178. I think of adding "System requirements" menu item in Luke's MenuBar. [~thetaphi], or someone else, if you have any comments about this (required java version) please let me know.
179. Hi [~Tomoko Uchida], I would not add this to the user interface. If somebody has started it, he will see that its not zoomed correctly. Anybody who used java probgrams on HiDPI display knows this already. The correct place to put the warning is into the SYSTEM_REQUIREMENTS.md/txt file in the Lucene root folder. It is then also part of official documentation ("ant documentation"). Please don't open any warning popups when starting! I am already annoyed by the always opening "open index" window. IMHO, the app should start without any popup windows. You can print a warning (using the Constants.JDK_IS_MINIMUM_JAVA11 field) in the log file.
180. Thanks for the comment, [~thetaphi]. I would like to add a notice, "For HiDPI graphics on Windows/Linux, Java 11 or grater is recommended to run the bundled GUI application Luke." to right after the first line ("Apache Lucene runs on Java 8 or greater.") in SYSTEM_REQUIREMENTS.txt. {quote}You can print a warning (using the Constants.JDK_IS_MINIMUM_JAVA11 field) in the log file. {quote} OK, I will print some message in the footer bar or log file, if there are APIs to detect the display resolution. Why I thought of adding notes to the user interface, users are not necessarily realize that the graphical problem relates to Java version they use. (Not long ago a user opened an issue titled "UI problem" in the github repo, even though I mentioned about this in the download page :))
181. Hi [~thetaphi], {quote} The correct place to put the warning is into the SYSTEM_REQUIREMENTS.md/txt file in the Lucene root folder. It is then also part of official documentation ("ant documentation"). {quote} I put a short note to SYSTEM_REQUIREMENTS.txt. Can you please double check it, if that's ok I will cherry-pick this to branch_8x. (I won't change the master branch.) diff: [https://github.com/apache/lucene-solr/commit/e448173d363f906fa243d8440086d1f0689307b6]
182. It's ok.
183. Commit 0d95f92facc77c03307606176e3244161b3507c0 in lucene-solr's branch refs/heads/branch_8x from Adrien Grand [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=0d95f92] LUCENE-2562: Don't randomly enable term vectors on fields.
184. Commit 793635eb0ed429b7e38f999fd9c4dc2d131530e0 in lucene-solr's branch refs/heads/master from Adrien Grand [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=793635e] LUCENE-2562: Don't randomly enable term vectors on fields.
185. There's been a handful of jenkins builds that have seen failing luke tests since this was committed â€“ most seem to have been addressed by the DocumentsImplTest tweak Adrien recently committed, but this CommitsImplTest failing seed still reproduces for me (on both branch_8x and master) ... {noformat} [junit4] Started J0 PID(6034@localhost). [junit4] Suite: org.apache.lucene.luke.models.commits.CommitsImplTest [junit4] 2> ERROR StatusLogger No Log4j 2 configuration file found. Using default configuration (logging only errors to the console), or user programmatically provided configurations. Set system property 'log4j2.debug' to show Log4j 2 internal initialization logging. See https://logging.apache.org/log4j/2.x/manual/configuration.html for instructions on how to configure Log4j 2 [junit4] 2> ERROR StatusLogger Could not reconfigure JMX [junit4] 2> java.security.AccessControlException: access denied ("javax.management.MBeanServerPermission" "createMBeanServer") [junit4] 2> at java.base/java.security.AccessControlContext.checkPermission(AccessControlContext.java:472) [junit4] 2> at java.base/java.security.AccessController.checkPermission(AccessController.java:895) [junit4] 2> at java.base/java.lang.SecurityManager.checkPermission(SecurityManager.java:322) [junit4] 2> at java.management/java.lang.management.ManagementFactory.getPlatformMBeanServer(ManagementFactory.java:479) [junit4] 2> at org.apache.logging.log4j.core.jmx.Server.reregisterMBeansAfterReconfigure(Server.java:140) [junit4] 2> at org.apache.logging.log4j.core.LoggerContext.setConfiguration(LoggerContext.java:559) [junit4] 2> at org.apache.logging.log4j.core.LoggerContext.reconfigure(LoggerContext.java:620) [junit4] 2> at org.apache.logging.log4j.core.LoggerContext.reconfigure(LoggerContext.java:637) [junit4] 2> at org.apache.logging.log4j.core.LoggerContext.start(LoggerContext.java:231) [junit4] 2> at org.apache.logging.log4j.core.impl.Log4jContextFactory.getContext(Log4jContextFactory.java:153) [junit4] 2> at org.apache.logging.log4j.core.impl.Log4jContextFactory.getContext(Log4jContextFactory.java:45) [junit4] 2> at org.apache.logging.log4j.LogManager.getContext(LogManager.java:194) [junit4] 2> at org.apache.logging.log4j.LogManager.getLogger(LogManager.java:581) [junit4] 2> at org.apache.lucene.luke.util.LoggerFactory.getLogger(LoggerFactory.java:70) [junit4] 2> at org.apache.lucene.luke.models.commits.CommitsImpl.<clinit>(CommitsImpl.java:44) [junit4] 2> at org.apache.lucene.luke.models.commits.CommitsImplTest.testGetSegmentAttributes(CommitsImplTest.java:153) [junit4] 2> at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) [junit4] 2> at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) [junit4] 2> at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) [junit4] 2> at java.base/java.lang.reflect.Method.invoke(Method.java:566) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner.invoke(RandomizedRunner.java:1750) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner$8.evaluate(RandomizedRunner.java:938) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner$9.evaluate(RandomizedRunner.java:974) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner$10.evaluate(RandomizedRunner.java:988) [junit4] 2> at org.apache.lucene.util.TestRuleSetupTeardownChained$1.evaluate(TestRuleSetupTeardownChained.java:49) [junit4] 2> at org.apache.lucene.util.AbstractBeforeAfterRule$1.evaluate(AbstractBeforeAfterRule.java:45) [junit4] 2> at org.apache.lucene.util.TestRuleThreadAndTestName$1.evaluate(TestRuleThreadAndTestName.java:48) [junit4] 2> at org.apache.lucene.util.TestRuleIgnoreAfterMaxFailures$1.evaluate(TestRuleIgnoreAfterMaxFailures.java:64) [junit4] 2> at org.apache.lucene.util.TestRuleMarkFailure$1.evaluate(TestRuleMarkFailure.java:47) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.StatementAdapter.evaluate(StatementAdapter.java:36) [junit4] 2> at

com.carrotsearch.randomizedtesting.ThreadLeakControl$StatementRunner.run(ThreadLeakControl.java:368) [junit4] 2> at com.carrotsearch.randomizedtesting.ThreadLeakControl.forkTimeoutingTask(ThreadLeakControl.java:817) [junit4] 2> at com.carrotsearch.randomizedtesting.ThreadLeakControl$3.evaluate(ThreadLeakControl.java:468) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner.runSingleTest(RandomizedRunner.java:947) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner$5.evaluate(RandomizedRunner.java:832) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner$6.evaluate(RandomizedRunner.java:883) [junit4] 2> at com.carrotsearch.randomizedtesting.RandomizedRunner$7.evaluate(RandomizedRunner.java:894) [junit4] 2> at org.apache.lucene.util.AbstractBeforeAfterRule$1.evaluate(AbstractBeforeAfterRule.java:45) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.StatementAdapter.evaluate(StatementAdapter.java:36) [junit4] 2> at org.apache.lucene.util.TestRuleStoreClassName$1.evaluate(TestRuleStoreClassName.java:41) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.NoShadowingOrOverridesOnMethodsRule$1.evaluate(NoShadowingOrOverridesOnMethodsRule.java:40) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.NoShadowingOrOverridesOnMethodsRule$1.evaluate(NoShadowingOrOverridesOnMethodsRule.java:40) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.StatementAdapter.evaluate(StatementAdapter.java:36) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.StatementAdapter.evaluate(StatementAdapter.java:36) [junit4] 2> at org.apache.lucene.util.TestRuleAssertionsRequired$1.evaluate(TestRuleAssertionsRequired.java:53) [junit4] 2> at org.apache.lucene.util.TestRuleMarkFailure$1.evaluate(TestRuleMarkFailure.java:47) [junit4] 2> at org.apache.lucene.util.TestRuleIgnoreAfterMaxFailures$1.evaluate(TestRuleIgnoreAfterMaxFailures.java:64) [junit4] 2> at org.apache.lucene.util.TestRuleIgnoreTestSuites$1.evaluate(TestRuleIgnoreTestSuites.java:54) [junit4] 2> at com.carrotsearch.randomizedtesting.rules.StatementAdapter.evaluate(StatementAdapter.java:36) [junit4] 2> at com.carrotsearch.randomizedtesting.ThreadLeakControl$StatementRunner.run(ThreadLeakControl.java:368) [junit4] 2> at java.base/java.lang.Thread.run(Thread.java:834) [junit4] 2> NOTE: reproduce with: ant test -Dtestcase=CommitsImplTest -Dtests.method=testGetSegmentAttributes -Dtests.seed=1D1E88937E689247 -Dtests.slow=true -Dtests.badapples=true -Dtests.locale=it -Dtests.timezone=US/Pacific-New -Dtests.asserts=true -Dtests.file.encoding=UTF-8 [junit4] FAILURE 0.72s | CommitsImplTest.testGetSegmentAttributes <<< [junit4] > Throwable #1: java.lang.AssertionError [junit4] > at __randomizedtesting.SeedInfo.seed([1D1E88937E689247:8686382357FBE3CA]:0) [junit4] > at org.apache.lucene.luke.models.commits.CommitsImplTest.testGetSegmentAttributes(CommitsImplTest.java:155) [junit4] > at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) [junit4] > at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) [junit4] > at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) [junit4] > at java.base/java.lang.reflect.Method.invoke(Method.java:566) [junit4] > at java.base/java.lang.Thread.run(Thread.java:834) [junit4] 2> NOTE: leaving temporary files on disk at: /home/hossman/lucene/alt_dev/lucene/build/luke/test/J0/temp/lucene.luke.models.commits.CommitsImplTest_1D1E88937E689247-001 [junit4] 2> NOTE: test params are: codec=SimpleText, sim=Asserting(org.apache.lucene.search.similarities.AssertingSimilarity@512d39d5), locale=it, timezone=US/Pacific-New [junit4] 2> NOTE: Linux 3.19.0-84-generic amd64/Oracle Corporation 11.0.2 (64-bit)/cpus=4,threads=1,free=238595688,total=260046848 [junit4] 2> NOTE: All tests run in this JVM: [CommitsImplTest] [junit4] Completed [1/1 (1!)] in 1.32s, 1 test, 1 failure <<< FAILURES! [junit4] [junit4] [junit4] Tests with failures [seed: 1D1E88937E689247]: [junit4] - org.apache.lucene.luke.models.commits.CommitsImplTest.testGetSegmentAttributes {noformat}

186. [~hossman]: Thanks for the report. Seems this is not related to particular seeds, but for example: {code} ant -f lucene/luke/build.xml test -Dtestcase=CommitsImplTest -Dtests.seed=998C8838349419D6 {code} As a starting point, the error is suppressed for me by adding those JMX permissions to the policy file. {code:java} permission javax.management.MBeanServerPermission "createMBeanServer"; permission javax.management.MBeanPermission "org.apache.logging.log4j.core.jmx.LoggerContextAdmin#-[*:*]", "instantiate,registerMBean,unregisterMBean"; permission javax.management.MBeanPermission "org.apache.logging.log4j.core.jmx.LoggerContextAdmin#*[*:*]", "queryNames"; permission "javax.management.MBeanTrustPermission" "register"; {code} [~thetaphi]: Do you have any suggestions or intuition about this?
187. I'm pretty sure the AccessControlException / javax.management.MBeanServerPermission error messages are unrealted to the test failure ... i see them loged by luke tests even when the test passs .. while they are probably worthdealining with, the real falure i was trying to draw your attention to was the assertion failure... {noformat} [junit4] FAILURE 0.72s | CommitsImplTest.testGetSegmentAttributes <<< [junit4] > Throwable #1: java.lang.AssertionError [junit4] > at __randomizedtesting.SeedInfo.seed([1D1E88937E689247:8686382357FBE3CA]:0) [junit4] > at org.apache.lucene.luke.models.commits.CommitsImplTest.testGetSegmentAttributes(CommitsImplTest.java:155) {noformat}
188. Commit 7ba8bdca8fac64826e8d9a2e4db5b779a2fa6f83 in lucene-solr's branch refs/heads/master from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=7ba8bdc] LUCENE-2562: Suppress SimpleText codec for CommitsImplTest.
189. Commit 116ec6d780f2b0498c466b926a306e7b7074035e in lucene-solr's branch refs/heads/branch_8x from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=116ec6d] LUCENE-2562: Suppress SimpleText codec for CommitsImplTest.
190. I found the failure comes from {{SimpleText}} codec (at least as far as the seed: 1D1E88937E689247). I think this is a codec for testing or debugging so I simply suppressed it for the failed test.
191. Commit 511efc89fe2832b05fcaa36e8759b3930c0fa9f1 in lucene-solr's branch refs/heads/master from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=511efc8] LUCENE-2562: Use Map#copyOf(), List#copyOf() and Set#copyOf() instead of Collections#unmodifiableMap(), unmodifiableList() and unmodifiableSet()
192. Commit 62ed046e52df0d120c285819c7b9680288681b0b in lucene-solr's branch refs/heads/master from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=62ed046] LUCENE-2562: Preserve map key order for the user interface.
193. Commit e960a4ab813c78ee9ce9223ced53a990ac66f999 in lucene-solr's branch refs/heads/branch_8x from Tomoko Uchida [https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=e960a4a] LUCENE-2562: Preserve map key order for the user interface.
194. [~Tomoko Uchida] Should we close this Jira again?
195. Seems tests are stable for now, I just closed this. Please reopen if needed.