

git_comments:

1. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
2. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
3. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
4. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
5. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
6. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
7. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
8. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
9. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
10. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
11. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
12. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
13. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
14. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
15. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
16. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
17. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
18. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
19. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
20. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
21. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally
22. See: <https://issues.apache.org/jira/browse/SOLR-12028> Tests cannot remove files on Windows machines occasionally

git_commits:

1. **summary:** SOLR-12028: BadApple and AwaitsFix annotations usage
message: SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit ad7e94a)

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

1. **summary:** BadApple and AwaitsFix annotations usage
description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.
label: test
2. **summary:** BadApple and AwaitsFix annotations usage
description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.
3. **summary:** BadApple and AwaitsFix annotations usage
description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.
4. **summary:** BadApple and AwaitsFix annotations usage
description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.
5. **summary:** BadApple and AwaitsFix annotations usage
description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests

15. summary: BadApple and AwaitsFix annotations usage

16. summary: BadApple and AwaitsFix annotations usage

17. summary: BadApple and AwaitsFix annotations usage

18. summary: BadApple and AwaitsFix annotations usage

19. summary: BadApple and AwaitsFix annotations usage

20. summary: BadApple and AwaitsFix annotations usage

21. summary: BadApple and AwaitsFix annotations usage

label: documentation

22. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.

23. summary: BadApple and AwaitsFix annotations usage

label: documentation

24. summary: BadApple and AwaitsFix annotations usage

label: code-design

25. summary: BadApple and AwaitsFix annotations usage

26. summary: BadApple and AwaitsFix annotations usage

27. summary: BadApple and AwaitsFix annotations usage

28. summary: BadApple and AwaitsFix annotations usage

29. summary: BadApple and AwaitsFix annotations usage

30. **summary:** BadApple and AwaitsFix annotations usage

31. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests

40. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require immediate attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and

49. **summary:** BadApple and AwaitsFix annotations usage

50. **summary:** BadApple and AwaitsFix annotations usage

51. **summary:** BadApple and AwaitsFix annotations usage

52. summary: BadApple and AwaitsFix annotations usage

53. **summary:** BadApple and AwaitsFix annotations usage

54. **summary:** BadApple and AwaitsFix annotations usage

55. **summary:** BadApple and AwaitsFix annotations usage

56. **summary:** BadApple and AwaitsFix annotations usage

57. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `immediate` attention. The default for developers is that

[illegible]

[illegible]

76. summary: BadApple and AwaitsFix annotations usage

77. summary: BadApple and AwaitsFix annotations usage

78. summary: BadApple and AwaitsFix annotations usage

79. **summary:** BadApple and AwaitsFix annotations usage

80. **summary:** BadApple and AwaitsFix annotations usage

81. summary: BadApple and AwaitsFix annotations usage

82. summary: BadApple and AwaitsFix annotations usage

83. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitFix. This is, of course, a judgement call - AwaitFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and

84. **summary:** BadApple and AwaitsFix annotations usage

85. **summary:** BadApple and AwaitsFix annotations usage

86. **summary:** BadApple and AwaitsFix annotations usage

87. **summary:** BadApple and AwaitsFix annotations usage

88. **summary:** BadApple and AwaitsFix annotations usage

89. **summary:** BadApple and AwaitsFix annotations usage

90. **summary:** BadApple and AwaitsFix annotations usage

91. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at S

92. **summary:** BadApple and AwaitsFix annotations usage

[illegible]

[illegible]

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.

- [illegible]

they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.

119. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.

120. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.

121. **summary:** BadApple and AwaitsFix annotations usage

description: There's a long discussion of this topic at SOLR-12016. Here's a summary: - BadApple annotations are used for tests that intermittently fail, say < 30% of the time. Tests that fail more often should be moved to AwaitsFix. This is, of course, a judgement call - AwaitsFix annotations are used for tests that, for some reason, the problem can't be fixed immediately. Likely reasons are third-party dependencies, extreme difficulty tracking down, dependency on another JIRA etc. Jenkins jobs will typically run with BadApple disabled to cut down on noise. Periodically Jenkins jobs will be run with BadApples enabled so BadApple tests won't be lost and reports can be generated. Tests that run with BadApples disabled that fail require `_immediate_` attention. The default for developers is that BadApple is enabled. If you are working on one of these tests and cannot get the test to fail locally, it is perfectly acceptable to comment the annotation out. You should let the dev list know that this is deliberate. This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. I've assigned this to myself to track so I don't lose track of it. No one person will fix all of these issues, this will be an ongoing technical debt cleanup effort.

jira_issues_comments:

1. **body:** Do we really need to close the individual test issues as well? That lets us link to the individual issue for a test in the annotation and keep discussion and history related to each test separate as necessary. We can still have an overall tracking issue like this, but it doesn't seem like a great substitute for having issues individually for bad tests.
label: test
2. [~markrmiller@gmail.com] Yeah, as I was going through the linked issues I realized that ones with discussion were valuable. So I left the ones that had any discussion or investigation info open and didn't change the annotation (except perhaps to change from AwaitsFix to BadApple). The ones I closed had no information/investigation info, they were just "BadApple because it fails too often"...
3. The attached patch is what it's looking like so far. Below is a list of all the BadApple and AwaitsFix annotations in the code base currently. I have to go over this again and run the test suite, but wanted to post them in case people want to keep some running because they're actively getting attention. Some of the failing tests are in areas getting a lot of active work so may be good candidates for leaving in, just let me know. Assuming that tests pass, I'm probably going to check these in Monday sometime after making another pass to see whether I put them in the right category. I intend to post the below info to the dev list each Sunday (or so) so these tests don't get lost forever. AtomicUpdateProcessorFactoryTest.java testMultipleThreads() @BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-10734") AutoAddReplicasIntegrationTest.java testSimple() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") AutoScalingHandlerTest.java testReadApi() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") AutoscalingHistoryHandlerTest.java testHistory() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") CdcBootstrapTest.java testBootstrapWithContinuousIndexingOnSourceCluster() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") CdcReplicationDistributedZkTest.java distribSetUp() @BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-12028") ComputePlanActionTest.java testSelectedCollections() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") DocValuesNotIndexedTest.java testGroupingDocAbsent() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") ForceLeaderTest.java testReplicasInLIRNoLeader() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") HdfsChaosMonkeyNothingIsSafeTest.java void @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") HdfsDirectoryTest.java testEOF() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") HttpPartitionTest.java test() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") LeaderFailoverAfterPartitionTest.java test() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") LeaderInitiatedRecoveryOnShardRestartTest.java void @LuceneTestCase.BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-10071") MoveReplicaHDFSTest.java testNormalFailedMove() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") MoveReplicaTest.java testFailedMove() // @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/SOLR-11458") PeerSyncReplicationTest.java test() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") ReplaceNodeNoTargetTest.java test() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") ReplicationFactorTest.java test() @LuceneTestCase.BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-6944") SSLMigrationTest.java test() @BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-6213") ShardSplitTest.java test() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") SharedFSAutoReplicaFailoverTest.java test() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") StreamExpressionTest.java testDistributions() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestAuthenticationFramework.java testBasics() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestCloudPseudoReturnFields.java testMultiValuedRTG() // @AwaitsFix(ed) TestCollapseQParserPlugin.java testStringCollapse() @AwaitsFix(bugUrl="https://issues.apache.org/jira/browse/SOLR-11974") TestControlledRealTimeReopenThread.java testCRTReopen() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-5737") TestDelegationWithHadoopAuth.java testDelegationTokenCancelFail() // @AwaitsFix(bugUrl="https://issues.apache.org/jira/browse/HADOOP-14044") TestGeo3DPoint.java testRandomBig() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestCUNormalizer2CharFilter.java testRandomStrings() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-5595") TestICUTokenizerCJK.java testRandomHugeStrings() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-5575") TestICUTokenizerCJK.java testRandomStrings() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-5575") TestImpersonationWithHadoopAuth.java testForwarding() @AwaitsFix(bugUrl="https://issues.apache.org/jira/browse/HADOOP-9893") TestJmxIntegration.java testJmxOnCoreReload() @Test @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-2715") // timing problem? TestLTRReRankingPipeline.java testDifferentTopN() @BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-12028") TestLargeCluster.java testSearchRate() @BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-11714") TestManagedResourceStorage.java testZkBasedJsonStorage() @LuceneTestCase.BadApple(bugUrl = "https://issues.apache.org/jira/browse/SOLR-6443") TestMinMaxOnMultiValuedField.java testDoubleFieldCache() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-6709") TestMinMaxOnMultiValuedField.java testFloatFieldCache() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-6709") TestMinMaxOnMultiValuedField.java testIntFieldCache() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-6709") TestMinMaxOnMultiValuedField.java testLongFieldCache() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-6709") TestMoreLikeThis.java testMultiFieldShouldReturnPerFieldBooleanQuery() @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-7161") TestPrepRecovery.java testLeaderNotResponding() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestPrepRecovery.java testLeaderUnloaded() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestPullReplica.java testAddDocs() @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestReplicationHandler.java doTestIndexAndConfigReplication()

- @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestReplicationHandler.java doTestIndexFetchOnMasterRestart()
// @AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/SOLR-9036") TestSegmentSorting.java testSegmentTerminateEarly()
// @Test @BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestSolrCloudWithDelegationTokens.java testDelegationTokenCancelFail()
// @AwaitsFix(bugUrl="https://issues.apache.org/jira/browse/HADOOP-14044") TestStressCloudBlindAtomicUpdates.java test_dv_stored()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestTlogReplica.java testRecovery()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestTriggerIntegration.java testCooldown()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestTriggerIntegration.java testListeners()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestTriggerIntegration.java testSearchRate()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TestUtilizeNode.java test()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TriggerIntegrationTest.java testEventQueue()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") TriggerIntegrationTest.java testMetricTrigger()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028") UIMABaseAnalyzerTest.java testRandomStrings() @Test @AwaitsFix(bugUrl =
"https://issues.apache.org/jira/browse/LUCENE-3869") UIMABaseAnalyzerTest.java testRandomStringsWithConfigurationParameters() @Test
@AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-3869") UIMATypeAwareAnalyzerTest.java testRandomStrings() @Test
@AwaitsFix(bugUrl = "https://issues.apache.org/jira/browse/LUCENE-3869") ZkControllerTest.java testPublishAndWaitForDownStates() @BadApple(bugUrl =
"https://issues.apache.org/jira/browse/SOLR-12028") ZkSolrClientTest.java testMultipleWatchesAsync()
@BadApple(bugUrl="https://issues.apache.org/jira/browse/SOLR-12028")
4. Attaching build system patch since SOLR-12016 was deleted for some reason.
 5. I think this is ready to check in, I'll do that later today along with Uwe's changes to the build system. From there we can refine how we work on the backlog.
 6. Commit 1fe4560b91b368b5fcd20e3c86e401ab4f9c6a6 in lucene-solr's branch refs/heads/master from Erick Erickson [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=1fe4560] SOLR-12028: BadApple and AwaitsFix annotations usage
 7. Commit 7a9b011d3eaa4088257e5724e12e8d018923c19 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=7a9b011] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 1fe4560)
 8. Commit 3b6307368e16894d58a2b57c8b15674c6239a243 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=3b63073] SOLR-12028: BadApple and AwaitsFix annotations usage. Removing of AwaitsFix on master never merged into 7x branch, SOLR-10136
 9. OK, we'll see how this goes. I'm going to leave this JIRA open for the nonce, once tests are settled I'll close it and we can move on.
 10. From [http://jenkins.sarowe.net/job/Lucene-Solr-reproduce-failed-tests/1760/], reproducing failures on [http://jenkins.sarowe.net/job/Lucene-Solr-tests-master/15259/]: {noformat} [repro] Failures at the tip of master without a seed: [repro] 5/5 failed: org.apache.solr.cloud.SSLMigrationTest [repro] 5/5 failed: org.apache.solr.cloud.ZkControllerTest [repro] 5/5 failed: org.apache.solr.core.TestJmxIntegration [repro] 5/5 failed: org.apache.solr.handler.TestReplicationHandler [repro] 5/5 failed: org.apache.solr.ltr.TestLTRReRankingPipeline {noformat} From [https://builds.apache.org/job/Lucene-Solr-repro/154/], reproducing failures on [https://builds.apache.org/job/Lucene-Solr-Tests-7.x/464/]: {noformat} [repro] Failures at the tip of branch_7x without a seed: [repro] 5/5 failed: org.apache.solr.cloud.SSLMigrationTest [repro] 5/5 failed: org.apache.solr.cloud.ZkControllerTest [repro] 5/5 failed: org.apache.solr.core.TestJmxIntegration [repro] 5/5 failed: org.apache.solr.handler.TestReplicationHandler [repro] 5/5 failed: org.apache.solr.ltr.TestLTRReRankingPipeline {noformat}
 11. [~steve_rowe] Right, but this is expected since tests.badapples=true is the default. I'm setting up a mail filter to funnel any report that does _not_ have tests.badapples=false to a special folder. [~thetaphi] Are we going to set up some number of runs with badapples=false? Any of _those_ that fail need immediate attention.....
 12. One more, from [http://jenkins.sarowe.net/job/Lucene-Solr-reproduce-failed-tests/1772/], reproducing failures on [http://jenkins.sarowe.net/job/Lucene-Solr-tests-master/15265/]: {noformat} [repro] Failures at the tip of master without a seed: [...] [repro] 5/5 failed: org.apache.solr.update.processor.AtomicUpdateProcessorFactoryTest {noformat}
 13. bq. Steve Rowe Right, but this is expected since tests.badapples=true is the default. Um, clearly I haven't been following closely enough, but: a regime that guarantees failed Jenkins runs 100% of the time is the opposite of useful.
 14. bq. Uwe Schindler Are we going to set up some number of runs with badapples=false? Any of those that fail need immediate attention..... IMHO, the vast majority of Jenkins runs should be configured with badapples=false. I'll go change every one of the existing jobs right now. [~erickerickson], I think your question should be rephrased: "Once all Jenkins runs are configured badapples=false, are we going to set up some number of runs with badapples=true?"
 15. [~steve_rowe] Fair enough. I expect it to take a bit to make this smooth. Here's the goal as I see it, subject to negotiation. How we work it out is TBD: Tests that are always broken are AwaitsFix. These won't be run regularly. I'm going to provide a report each week (just an e-mail to the dev list) with the current tests with either AwaitsFix or BadApple annotations. BadApple tests should be run often enough to give various summaries something to report on (e.g. Hoss' report, your work). Personally I'm going to filter them to a special folder that I can examine at leisure. Anything run with tests.badapples=false that fails requires immediate attention. For a short while that may just be adding BadApple annotations. When things settle down, adding more BadApple annotations should only rarely be done.
 16. bq. BadApple tests should be run often enough to give various summaries something to report on I think I saw weekly frequency suggested elsewhere for badapple tests, but then Hoss's 7-day summary won't have much to work with. I think daily is a better frequency. I'll set up daily job with badapple=true for master and branch_7x, and another set of weekly jobs with nightly=true & badapples=true (there are a few tests with this annotation combination), on both Apache and my Jenkins. [~thetaphi], do you want to host badapple tests on Policeman Jenkins?
 17. bq. Uwe Schindler, do you want to host badapple tests on Policeman Jenkins? Yes, sure. I'd prefer {{@daily/2}}. When this is commit I will duplicate Master and 7.x Jobs. For release branches it makes no sense to run BadApple tests. But that's my opinion.
 18. [~steve_rowe], with this patch you don't need to change the existing jobs. If you run "ant jenkins-hourly" ant task it will not run bad apples. If you run "jenkins-badapples" it's the new variant. It's all in build file!
 19. [~thetaphi] This is committed, I'm keeping it open for a bit while we adjust.
 20. Steve: see here: https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;a=blobdiff;f=build.xml;h=ceff8d2a36b6a30a5ffff2b517a8b32ed6e31153;hp=a5f09e48711447852e20cad1c6f98c270abd0193;hb=1fe4560;hpb=1fc3ca0cbb2f
 21. **body:** Commit 98d85eadd4ed0ebab3e4c37aff88749350934524 in lucene-solr's branch refs/heads/master from [~thetaphi] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=98d85ea] SOLR-12028: Fix typo in property name
label: documentation
 22. **body:** Commit 73d5b070c911d03264a600521df06b2b07466304 in lucene-solr's branch refs/heads/branch_7x from [~thetaphi] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=73d5b07] SOLR-12028: Fix typo in property name
label: documentation
 23. **body:** Hi [~steve_rowe]: There was a typo in property name, the ANT tasks are setup correctly. I removed the useless property from jenkins, it should just run "ant jenkins-hourly" or "ant jenkins-badapples", nothing more. I did not touch ASF Jenkins. I will setup daily jobs calling "ant jenkins-badapples" on policeman later!
label: documentation
 24. **body:** Thanks [~thetaphi], I'll remove the useless property from ASF Jenkins jobs.
label: code-design
 25. Commit 7f72a4a5262aaf9276d8f3b17c56ac35abe6d48f in lucene-solr's branch refs/heads/branch_7x from [~steve_rowe] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=7f72a4a] SOLR-12028: add jenkins-nightly-badapples target
 26. Commit 939aedfed7a66065e2dfa42efb6fd4b53f9fab12 in lucene-solr's branch refs/heads/master from [~steve_rowe] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=939aedf] SOLR-12028: add jenkins-nightly-badapples target
 27. Good idea with the extra target.
 28. Hi, I just noticed that the reproduce line was not consistent with all test groups. Also the sysprop defaults were initialized differently to the underlying annotations. This patch fixes this and makes the defaults "dynamic" (from annotation): [^SOLR-12028-sysprops-reproduce.patch] Will commit in a moment.
 29. Commit 7dba350c7a02fe603faec49227ff2672e4d8e6ae in lucene-solr's branch refs/heads/master from [~thetaphi] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=7dba350] SOLR-12028: Make initialization of constants dynamic (by reading the annotation), also add missing reproduce info
 30. Commit 9b7e35e69b694d9006c0964ae7a26ed2694ea055 in lucene-solr's branch refs/heads/branch_7x from [~thetaphi] [https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=9b7e35e] SOLR-12028: Make initialization of constants dynamic (by reading the annotation), also add missing reproduce info
 31. Last run had following test failures, although {{tests.badapples=false}}: - org.apache.solr.cloud.autoscaling.TriggerIntegrationTest.testNodeMarkersRegistration - org.apache.solr.cloud.autoscaling.TriggerIntegrationTest.testSearchRate - org.apache.solr.cloud.hdfs.HdfsRecoveryZkTest

32. Commit b732f067751afc7074bfff1284c1f862b078a9b6c in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=b732f06>] SOLR-12028: BadApple and AwaitsFix annotations usage
33. Commit 9363529f4bffc98be3955e2e0e0229eebeb43a5a in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=9363529>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit b732f06)
34. I've very confused by some of the changes made under the banner of this issue, and since SOLR-12016 was evidently deleted, it's hard to find the "long discussion of this topic" to see if it's explained there (i could have tried to piece it together from mail archives, but instead i'm just going to take the summary in *this* jira at face value) ... I draw attention to this comment... {quote} This JIRA is a placeholder for BadApple tests to point to between the times they're identified as BadApple and they're either fixed or changed to AwaitsFix or assigned their own JIRA. {quote} ...and yet, when i look at {{ZkControllerTest.testPublishAndWaitForDownStates}} I can not wrap my head around the changes made to that test as part of this issue. Prior to the creation of SOLR-12028... * {{ZkControllerTest.testPublishAndWaitForDownStates}} was disabled using {{AwaitsFix}} * The {{AwaitsFix}} annotation pointed to SOLR-7736 which was an OPEN issue because the test was known to be broken ** admittedly, the issue didn't have much in the description, but it was explicitly open knowing that this test was currently broken * As a result, this completely broken test was never being run and never failing. As a result of SOLR-12028... * SOLR-7736 is now marked "Resolved:Duplicate" ** even though there is no "Duplications" link to any other jira ????? * The {{AwaitsFix}} annotation was replaced by a {{BadApple}} annotation which links here to SOLR-12028 *and* to SOLR-7736... ** even though the guidelines of both annotations (as i've always understood them) are that they aren't suppose to link to closed issues??? * so now this test is running for all developers, even though it was known in advance to be broken and "awaiting fix" In short: this test already met the "end goal" of "either fixed or changed to AwaitsFix or assigned their own JIRA." and yet now it's in a worse state then when we started... *WAT?* What am i not understanding about this process? what is the point of mucking with tests like these that were already {{AwaitsFix}} pointing to *OPEN* issues???
35. Well, the process is evolving. I saw no use in keeping a separate open JIRA that simply said "this test doesn't succeed very often". I thought that pointing people at the uber-JIRA would provide more context. Whether that's the right call or not is an open question. SOLR-7736: Right, I linked it under "related to" rather than "duplicates", we can add a duplicates link in if you'd like. I put in both JIRAs to allow a complete record, if it's too confusing we can remove, say, the 7736 reference. As for AwaitsFix, the theory (which I may not have adhered to completely, feel free to correct) is that when something is actually known to be broken, i.e. code we can point to and say "this code is broken" doesn't need to be run until, well, the code is fixed. This would include dependent libs, code where the root cause is known but not fixed and the like. AwaitsFix aren't run periodically so disappear from sight. BadApple, OTOH, is "this test doesn't run reliably, we have no clue why". These are also run periodically for the various reports to be able to see. That way they don't get lost. It looked to me like AwaitsFix and BadApple were applied somewhat interchangeably so this is an attempt to bring regularity to the annotations. I reconstructed SOLR-12016 under SOLR-12037 BTW. bq: so now this test is running for all developers As it should IMO unless and until we do one of the following 1> decide it's just a crappy test and remove it 2> identify the root problem is and move it to AwaitsFix and link it to a JIRA with the RCA. 3> fix it and remove the annotation altogether. Note that for <3>, I'm starting by commenting out the annotation and providing the date it was commented out on the theory that if the test starts failing again it would be good information to have. OTOH if we look at this in 6 months and it hasn't failed in that time, the commented-out annotation can be removed. My two main goals are: 1> surface any tests that fail regularly with no RCA for reports etc. 2> get clean runs with BadApple=false to catch regressions I don't particularly care what the mechanics are. You'll see, every Sunday, a report to the dev list for the current BadApple and AwaitsFix annotations, file and method. This should complement your reports (which are way cool BTW). So that's all the theory, I expect a few things to shift around before it all settles out...
36. **body:** bq. As for AwaitsFix, the theory (which I may not have adhered to completely, feel free to correct) is that when something is actually known to be broken, i.e. code we can point to and say "this code is broken" doesn't need to be run until, well, the code is fixed. ... isn't that exactly the situation SOLR-7736 was in before you started this? the test was known to be broken and reliably fails 100% of the time, and had an open issue to track fixing it in the future? {quote} bq: so now this test is running for all developers As it should IMO unless and until we do one of the following ... 2> identify the root problem is and move it to AwaitsFix and link it to a JIRA with the RCA. ... {quote} As i said: SOLR-7736 already had a jira -- the description was sparse, but FWIW: it did link back to another jira that explained the problem with the test ... AFAICT it already met the "end goal" you were aiming for. ---- Bottom line... * I understood the initial motivation to: identify flaky tests; mark them as BadApple to get them off the regular jenkins builds; maybe mark them as AwaitsFix. * I also understand the value in saying "AwaitsFix pointed at a closed issue is useless" -- let's assess those, maybe try them as BadApple, open new issues / re-AwaitsFix if needed * I do not understand what happened with this test... ** why should *ANY* test that was _already_ marked AwaitsFix, pointing to an *open* jira have been changed to BadApple? ** why should any _open_ jira saying "this test is broken" have been resolved just to redirect here / open a new issue later? if the concern was that some issues didn't do a "good enough" job of explaining why the test was broken, then that should have been raised in those jiras. it just doesn't make much sense to me at all to resolve them (to just open new ones later). Like wise it seems completely bizarre to me to take a test that's already marked AwaitsFix, which can easily be demonstrated to fail 100% of the time and change it to "BadApple" so that it starts getting in the way and failing for all devs.
- label:** test
37. Then change it to point back to 7736, I really don't care. My initial changes were essentially straw-man. I'm perfectly happy to have that late-night judgement call reversed if you feel strongly about it. Here's the tests that have failed since 12028 (not even including the Lucene "can't delete file" list), we have far more productive things to do than argue over that one. junit.framework.TestSuite.org.apache.solr.cloud.TestSolrCloudWithSecureImpersonation org.apache.lucene.analysis.core.TestRandomChains.org.apache.lucene.analysis.core.TestRandomChains.testRandomChains org.apache.solr.client.solrj.request.TestV2Request.testCloudSolrClient org.apache.solr.client.solrj.request.TestV2Request.testHttpSolrClient org.apache.solr.cloud.AliasIntegrationTest.testMetadata org.apache.solr.cloud.AliasIntegrationTest.testModifyMetadataCAR org.apache.solr.cloud.ChaosMonkeySafeLeaderTest.test org.apache.solr.cloud.CollectionsAPIDistributedZkTest.testCollectionsAPI org.apache.solr.cloud.LegacyCloudClusterPropTest.testCreateCollectionSwitchLegacyCloud org.apache.solr.cloud.ShardSplitTest.testSplitAfterFailedSplit org.apache.solr.cloud.ShardSplitTest.testSplitWithChaosMonkey org.apache.solr.cloud.SharedFSAutoReplicaFailoverTest.test org.apache.solr.cloud.TestStressCloudBlindAtomicUpdates.test_dv org.apache.solr.cloud.ZkShardTermsTest.testParticipationOfReplicas org.apache.solr.cloud.api.collections.CollectionsAPIDistributedZkTest org.apache.solr.cloud.api.collections.CollectionsAPIDistributedZkTest.testCollectionsAPI org.apache.solr.cloud.api.collections.ShardSplitTest.testSplitAfterFailedSplit org.apache.solr.cloud.api.collections.ShardSplitTest.testSplitWithChaosMonkey org.apache.solr.cloud.api.collections.TestCollectionsAPIViaSolrCloudCluster.testCollectionCreateSearchDelete org.apache.solr.cloud.autoscaling.sim.TestTriggerIntegration.testNodeLostTriggerRestoreState org.apache.solr.core.TestDynamicLoading.testDynamicLoading
38. bq. why should any open jira saying "this test is broken" have been resolved just to redirect here / open a new issue later? As I mentioned above, I also don't think this is the way to go. It's fine to open an issue to track all this work, but makes no sense to close the issues per problem test. bq. why should ANY test that was already marked AwaitsFix, pointing to an open jira have been changed to BadApple? The only reason I believe that should happen is because the test does not fail very often vs failing a lot. bq. we have far more productive things to do than argue over that one. I don't think any of us are interested in arguing, just taking the best approach. Our annotations are made to point at a single issue where we can discuss what is wrong with a test and attach patches, etc. When it's closed we can remove the annotation. Closing those kinds of issues for this bulk issue is not something I want to argue about - it's something that seems ... wrong. This issue should be about annotations. Let the other issues live and be about fixing the individual tests. They close when the annotations go away.
39. +1 at [~markrmiller@gmail.com]
40. Tests I'm about to BadApple
41. Commit ad7e94afb25ba72832f5bc887f1cb4c3e5040622 in lucene-solr's branch refs/heads/master from Erick [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=ad7e94a>] SOLR-12028: BadApple and AwaitsFix annotations usage
42. Commit 42d1abba53233ac51130e0cd8df9a80e07d92ce1 in lucene-solr's branch refs/heads/branch_7x from Erick [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=42d1abb>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit ad7e94a)
43. Commit 23765a9ff952e48a7892e5b140ec4c645f2de7b6 in lucene-solr's branch refs/heads/master from [~tflobbe] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=23765a9>] SOLR-12028, SOLR-7736: testPublishAndWaitForDownStates always fails
44. Commit dc9b511f7815c8a95e88d55e523e9c28e3299a3a in lucene-solr's branch refs/heads/branch_7x from [~tflobbe] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=dc9b511>] SOLR-12028, SOLR-7736: testPublishAndWaitForDownStates always fails
45. {{testPublishAndWaitForDownStates}} was failing consistently to me too. I changed it to {{@AwaitsFix}}
46. Commit f1f929b15a4bc0876e8c241bbc184fa870662939 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=f1f929b>] SOLR-12028: BadApple and AwaitsFix annotations usage
47. Commit 037f1ee3b8afe78c510211a228203f18bfe84220 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=037f1ee>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit f1f929b)
48. Commit a9fd0754ba1a9e6349f4216abf3b46ae8ea320c3 in lucene-solr's branch refs/heads/master from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=a9fd075>] SOLR-12028: Remove BadApples for LIRRollingUpdatesTest
49. Commit 2f2be43170cf973eee771646f7729a89c95f00df in lucene-solr's branch refs/heads/branch_7x from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=2f2be43>] SOLR-12028: Remove BadApples for LIRRollingUpdatesTest

50. Commit 0cd9f7f35af84ce769a37a2ef1afe8990c5b061a in lucene-solr's branch refs/heads/master from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=0cd9f7f1>] SOLR-12028: Remove BadApples for ZkShardTermsTest.testParticipationOfReplicas
51. Commit b2856a14620aa3132899dcac3512e67162400c5b in lucene-solr's branch refs/heads/branch_7x from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=b2856a1>] SOLR-12028: Remove BadApples for ZkShardTermsTest.testParticipationOfReplicas
52. Commit c48177b4b61b17c8fa5f03bae5bbdc55b38edef9 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=c48177b>] SOLR-12028: BadApple and AwaitsFix annotations usage
53. Commit 741b0685f0399afb2cf1b64950536f1ff2a46dfb in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=741b068>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit c48177b)
54. Commit 23707314dd7fa67c7dd089d8fb6c1bece4817408 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=2370731>] SOLR-12028: BadApple and AwaitsFix annotations usage
55. Commit 95e1903d06345f75bbaec7f34901c3199bc254ed5 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=95e1903>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 2370731)
56. Commit 19847b9bb87fc9d03f7b6bf8dcba7a02f943b8930 in lucene-solr's branch refs/heads/branch_7x from [~simonw] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=19847b9>] Fix TestLRUQueryCache#testDocValuesUpdatesDontBreakCache this test was flagged as BadApple and referred to SOLR-12028 The test stated clearly that the usage of newSearch(reader) is dangerous since it might add concurrency to the test. This commit respects this comment and removes all subsequent useage of newSearcher(...)
57. Commit 24f10c48dc56de71d72e8bbddcc6a7d9a13f5edf6 in lucene-solr's branch refs/heads/master from [~simonw] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=24f10c4>] Fix TestLRUQueryCache#testDocValuesUpdatesDontBreakCache this test was flagged as BadApple and referred to SOLR-12028 The test stated clearly that the usage of newSearch(reader) is dangerous since it might add concurrency to the test. This commit respects this comment and removes all subsequent useage of newSearcher(...)
58. Thanks Simon!
59. Commit 0014f3af88367961d8a7eb84a1a2333ecf66cb46 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=0014f3a>] SOLR-12028: BadApple and AwaitsFix annotations usage
60. Commit a27d866cde625ccce2ae71d0115a59293257f1ae in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=a27d866>] SOLR-12028: BadApple and AwaitsFix annotations usage
61. Commit 3028f3e9a337f9b8b0d021ecf3f2cb54217c566 in lucene-solr's branch refs/heads/master from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=3028f3e>] SOLR-12028: Remove BadApple for CollectionsAPIDistributedZkTest.testCollectionReload()
62. Commit c662b7d266a532030576e6e2be928ff1fbb9c439 in lucene-solr's branch refs/heads/branch_7x from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=c662b7d>] SOLR-12028: Remove BadApple for CollectionsAPIDistributedZkTest.testCollectionReload()
63. Commit 5ef43e900f8abe56cb9bba8ca1d050ec956f21 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=5ef43e9>] SOLR-12028: BadApple and AwaitsFix annotations usage
64. Commit 3d21fda4ce1c899f31b8f00e200eb1ac0d23d17b in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=3d21fda>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 5ef43e9)
65. Commit 89fc02a3b04aedf2803a46782b3a8b8e9b5c9a7b in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=89fc02a>] SOLR-12028: BadApple and AwaitsFix annotations usage
66. Commit 2558a06f3c8040d47de6606da2539532e4c67854 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=2558a06>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 89fc02a)
67. Commit 25ebe0e35fb48a581a773a968732992e78eaae1 in lucene-solr's branch refs/heads/master from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=25ebe0e>] SOLR-12028: Remove TestCloudConsistency BadApples after beasting
68. Commit 0f9e221ed450b6b02fbd903c5a13226b61a00 in lucene-solr's branch refs/heads/branch_7x from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=0f9e221>] SOLR-12028: Remove TestCloudConsistency BadApples after beasting
69. Commit 4a9a8397e458a5805c55fe494ba4b6de18233f90 in lucene-solr's branch refs/heads/master from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=4a9a839>] SOLR-12028: Remove BadApple for TestCloudRecovery
70. Commit 96f6c65e43445bbb0b77604c8c6550ca654b5de8 in lucene-solr's branch refs/heads/branch_7x from [~caomanhdat] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=96f6c65>] SOLR-12028: Remove BadApple for TestCloudRecovery
71. Commit fd929c1d601bd6e946489f51e7b3c0887a3392a6 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=fd929c1>] SOLR-12028: BadApple and AwaitsFix annotations usage
72. Commit 76966ae3adddd5d867c7571654c611830eb76fdf in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=76966ae>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit fd929c1)
73. Still not getting Git revisions for checked in code, master - 5cd8aa4ff90e43b6c1cdc9902aaea432efb1558e 7x - ae918f7274b8872029f70058ad996399648e3cf8
74. Commit 9a9c65a6ba3503fe5e505b1b7ae8cc43da3a271f in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=9a9c65a>] SOLR-12028: Forgot to check in last Thursday
75. Commit 5f619276359bc751d6e94b9f9259ec03b185d83a in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=5f61927>] SOLR-12028: Forgot to check in last Thursday (cherry picked from commit 9a9c65a)
76. Commit a09f3facfc5d9d096661a2c45458fc5b55a07819 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=a09f3fa>] SOLR-12028: BadApple and AwaitsFix annotations usage
77. Commit d1500e5b38c5d09844f75a15e607fc3e2193a2e1 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=d1500e5>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit a09f3fa)
78. Commit 1bea1da5dc43d3b392c5e363c3ad970e1df6d5fc in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=1bea1da>] SOLR-12028: BadApple and AwaitsFix annotations usage
79. Commit b086323f5c46698ed407ec9af1e1f080f76155ac in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=b086323>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 1bea1da)
80. Commit 6afd3d11929a75e3b3310638b32f4ed55da3ea6e in lucene-solr's branch refs/heads/master from Erick [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=6afd3d1>] SOLR-12028: BadApple and AwaitsFix annotations usage
81. Commit aa144fc69cc63641de1062a014911ad594143e71 in lucene-solr's branch refs/heads/branch_7x from Erick [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=aa144fc>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 6afd3d11929a75e3b3310638b32f4ed55da3ea6e)
82. Commit bff7a3bbb188fe08a9a24711bd7b3c54e32d36e8 in lucene-solr's branch refs/heads/master from Tomas Fernandez Lobbe [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=bff7a3b>] SOLR-12222, SOLR-12028: Remove @BaddApple annotation from TestDistributedSearch This test hasn't failed since the changes in SOLR-12222 were committed
83. Commit 6db3d9fdd0e955bf496ac7903d0093b16a7be593 in lucene-solr's branch refs/heads/branch_7x from Tomas Fernandez Lobbe [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=6db3d9f>] SOLR-12222, SOLR-12028: Remove @BaddApple annotation from TestDistributedSearch This test hasn't failed since the changes in SOLR-12222 were committed
84. Commit dcbd5323bbdc2456d1ef0cbe2ee8612b9d58ffe1 in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=dcbd532>] SOLR-12028: BadApple and AwaitsFix annotations usage
85. Commit 4f1c0aab1cd7775c61dfefab3dc588e8dcf65706 in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=4f1c0aa>] SOLR-12028: BadApple and AwaitsFix annotations usage
86. Commit 3818936f0472043b139b0c9c7cb6dbed63dd197 in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=3818936>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit dcbd5323bbdc2456d1ef0cbe2ee8612b9d58ffe1)
87. Commit 9f615fbae70be7d3ee145ad9cd453c67204cf8a8 in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=9f615fb>] SOLR-12028: BadApple and AwaitsFix annotations usage
88. Commit f42d29d8130cbd9ab88aa04d114553754928945e in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=f42d29d>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 9f615fbae70be7d3ee145ad9cd453c67204cf8a8)
89. Commit aa10cb7802ca2f2e0159a84c180193db43ca7926 in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=aa10cb7>] SOLR-12028: BadApple and AwaitsFix annotations usage

90. Commit b3daa6ce5f7bf74ef148a58e609af1c43a283f44 in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=b3daa6c>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit aa10cb7802ca2f2e0159a84c180193db43ca7926)
91. Commit aa10cb7802ca2f2e0159a84c180193db43ca7926 in lucene-solr's branch refs/heads/jira/http2 from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=aa10cb7>] SOLR-12028: BadApple and AwaitsFix annotations usage
92. Commit 0dc66c236d5f61caad96e36454b4b15fbde35720 in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=0dc66c2>] SOLR-12028: BadApple and AwaitsFix annotations usage
93. Commit de38f8168e822811728884310df7bfb9604e2a6e in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=de38f81>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 0dc66c236d5f61caad96e36454b4b15fbde35720)
94. Commit 0dc66c236d5f61caad96e36454b4b15fbde35720 in lucene-solr's branch refs/heads/jira/http2 from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=0dc66c2>] SOLR-12028: BadApple and AwaitsFix annotations usage
95. Commit 910a1c03124d31f940259682340fef402a42c9ae in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=910a1c0>] SOLR-12028: BadApple and AwaitsFix annotations usage
96. Commit 20123377479b3d0689f106ec53aa9e1148a258a in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=2012337>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 910a1c03124d31f940259682340fef402a42c9ae)
97. Commit dd088fb83eeb48c752cc69a5dd173aa1c5224bc9 in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=dd088fb>] SOLR-12028: BadApple and AwaitsFix annotations usage
98. Commit c5056d00c60b532c49a062c4de47129350078614 in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=c5056d0>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit dd088fb83eeb48c752cc69a5dd173aa1c5224bc9)
99. Commit 2b4717c6f4e729dfaf56ba2d6e5c64d9cc3fbbdc in lucene-solr's branch refs/heads/master from [~nknize] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=2b4717c>] SOLR-12028: Reduce test data set for TestLatLonPolygonShapeQueries.testRandomBig
100. Commit 9daeb1b01969b2da33d2ff52ffb4a83f5250fad0 in lucene-solr's branch refs/heads/branch_7x from [~nknize] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=9daeb1b>] SOLR-12028: Reduce test data set for TestLatLonPolygonShapeQueries.testRandomBig
101. Commit 886a2de03dc5fdb6399e53cd76104e26c57e1eef in lucene-solr's branch refs/heads/branch_7_5 from [~nknize] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=886a2de>] SOLR-12028: Reduce test data set for TestLatLonPolygonShapeQueries.testRandomBig
102. Commit 6c5df58b5a87c44626427dbf7968dfdd284b5eba in lucene-solr's branch refs/heads/master from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=6c5df58>] SOLR-12028: BadApple and AwaitsFix annotations usage
103. Commit 8dec795db41cfb3ca6eb86fb8890704e348de2c6 in lucene-solr's branch refs/heads/branch_7x from [~cp.erickson@gmail.com] [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=8dec795>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit 6c5df58b5a87c44626427dbf7968dfdd284b5eba)
104. Commit 24ae0d16cd5446f27309270f6534d4c19fb4fe43 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=24ae0d1>] SOLR-12028: Catching up with annotations after recent Solr test work
105. Commit 53ac8989e25d435f25ca1c78b05384b357e44590 in lucene-solr's branch refs/heads/branch_7x from Erick Erickson [<https://git-wip-us.apache.org/repos/asf?p=lucene-solr.git;h=53ac898>] SOLR-12028: Catching up with annotations after recent Solr test work (cherry picked from commit 24ae0d16cd5446f27309270f6534d4c19fb4fe43)
106. [~erickerickson], I'd like to consider removing BadApple for BasicAuthIntegrationTest, it's been tagged for a month. How can I check whether it still fails and for what reason?
107. Commit 3235c61e7775728eb3d1c014e5e3a5d1ea3a55eb in lucene-solr's branch refs/heads/master from Erick Erickson [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=3235c61>] SOLR-12028: BadApple and AwaitsFix annotations usage.
108. Commit 4428f53c91c03397e8d542a0017f2e43a26e2d7 in lucene-solr's branch refs/heads/branch_8x from Erick Erickson [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=4428f53>] SOLR-12028: BadApple and AwaitsFix annotations usage. (cherry picked from commit 3235c61e7775728eb3d1c014e5e3a5d1ea3a55eb)
109. Commit e04a28e1400514e104a867f5d6c76f8761ff348a in lucene-solr's branch refs/heads/master from Erick Erickson [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=e04a28e>] SOLR-12028: BadApple and AwaitsFix annotations usage
110. Commit d420382d041ea078598451e882abff94e3039611 in lucene-solr's branch refs/heads/branch_8x from Erick Erickson [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=d420382>] SOLR-12028: BadApple and AwaitsFix annotations usage (cherry picked from commit e04a28e1400514e104a867f5d6c76f8761ff348a)
111. FWIW: It's my intention to review all of the issues still annotated with {{@BadApple(SOLR-12028)}} to see how many of them are problematic, and if so why. For tracking purposes, I'll make sure to include mention of this Jira in any commits to either remove or change the annotation (ie: if i identify an underlying bug and promote to {{@AwaitsFix}})
112. Commit 3fb39b2960ae3b04d8f97a683a8c9daaef3f4696 in lucene-solr's branch refs/heads/master from Chris M. Hostetter [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=3fb39b2>] TestDistributedStatsComponentCardinality: Remove @BadApple anotation This test was originally anoted @BadApple ~2018-10-14 as pat of SOLR-12028. So far in 2019, it has failed in 62 jenkins-BadApple builds, but all prior to 2019-06-14. Since we no longer have logs from those older jenkins builds, it's hard to be certain how/why this test was failing, or why exactly it *stopped* failing -- but a likely suspect is that the root cause of the problem may have been fixed as part of the Jetty upgrade in SOLR-13541.
113. Commit e6dc96d4d8d09fa70947c94ce1388be52acc42bf in lucene-solr's branch refs/heads/branch_8x from Chris M. Hostetter [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=e6dc96d>] TestDistributedStatsComponentCardinality: Remove @BadApple anotation This test was originally anoted @BadApple ~2018-10-14 as pat of SOLR-12028. So far in 2019, it has failed in 62 jenkins-BadApple builds, but all prior to 2019-06-14. Since we no longer have logs from those older jenkins builds, it's hard to be certain how/why this test was failing, or why exactly it *stopped* failing -- but a likely suspect is that the root cause of the problem may have been fixed as part of the Jetty upgrade in SOLR-13541. (cherry picked from commit 3fb39b2960ae3b04d8f97a683a8c9daaef3f4696)
114. Commit 3fb39b2960ae3b04d8f97a683a8c9daaef3f4696 in lucene-solr's branch refs/heads/jira/SOLR-13452_gradle_8 from Chris M. Hostetter [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=3fb39b2>] TestDistributedStatsComponentCardinality: Remove @BadApple anotation This test was originally anoted @BadApple ~2018-10-14 as pat of SOLR-12028. So far in 2019, it has failed in 62 jenkins-BadApple builds, but all prior to 2019-06-14. Since we no longer have logs from those older jenkins builds, it's hard to be certain how/why this test was failing, or why exactly it *stopped* failing -- but a likely suspect is that the root cause of the problem may have been fixed as part of the Jetty upgrade in SOLR-13541.
115. Commit 1411aaee94d49f26c55272f3876a4261357467c8 in lucene-solr's branch refs/heads/master from Chris M. Hostetter [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=1411aae>] HdFsRecoveryZkTest & HdFsNNFailoverTest: Remove @BadApple anotation These tests were originally anoted @BadApple in early 2018 as pat of SOLR-12028. Neither test has failed since 2018-12-28. Since we no longer have logs from those older jenkins builds, it's hard to be certain how/why this test was failing, or why exactly it stopped failing â€œ but it's possible the underlying issues were addressed by general hardening of SolrCloud and the associated base test classes around the same time.
116. Commit cb72085ee8cc5fa9229424c101a744f758042153 in lucene-solr's branch refs/heads/branch_8x from Chris M. Hostetter [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=cb72085>] HdFsRecoveryZkTest & HdFsNNFailoverTest: Remove @BadApple anotation These tests were originally anoted @BadApple in early 2018 as pat of SOLR-12028. Neither test has failed since 2018-12-28. Since we no longer have logs from those older jenkins builds, it's hard to be certain how/why this test was failing, or why exactly it stopped failing â€œ but it's possible the underlying issues were addressed by general hardening of SolrCloud and the associated base test classes around the same time. (cherry picked from commit 1411aaee94d49f26c55272f3876a4261357467c8)
117. Commit 1ca7067a810578d4e246b5434b9cdcec7145d230 in lucene-solr's branch refs/heads/master from Erick Erickson [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=1ca7067>] SOLR-12028: BadApple and AwaitsFix annotations usage, Unannotated tests that haven't failed in a while
118. Commit 164ecce26e0db5408122054ff1acc914b00b5ef6 in lucene-solr's branch refs/heads/branch_8x from Erick Erickson [<https://gitbox.apache.org/repos/asf?p=lucene-solr.git;h=164ecce>] SOLR-12028: BadApple and AwaitsFix annotations usage, Unannotated tests that haven't failed in a while (cherry picked from commit 1ca7067a810578d4e246b5434b9cdcec7145d230)
119. I'm not changing annotations nearly as often as previously. Between that and SOLR-14636, this Jira has outlived it's usefulness. Going forward, we can open up a new one if necessary.