

**git\_comments:**

1. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
2. \* \* The contract of OpenStack Swift: only enabled if the test binding data is provided
3. insert the base features
4. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
5. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
6. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
7. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
8. \* Licensed to the Apache Software Foundation (ASF) under one \* or more contributor license agreements. See the NOTICE file \* distributed with this work for additional information \* regarding copyright ownership. The ASF licenses this file \* to you under the Apache License, Version 2.0 (the \* "License"); you may not use this file except in compliance \* with the License. You may obtain a copy of the License at \* \* <http://www.apache.org/licenses/LICENSE-2.0> \* \* Unless required by applicable law or agreed to in writing, software \* distributed under the License is distributed on an "AS IS" BASIS, \* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \* See the License for the specific language governing permissions and \* limitations under the License.
9. \* \* root dir operations against an S3 bucket
10. Openstack Swift is a blobstore, with very different behavior than a classic filesystem.
11. ~ Licensed to the Apache Software Foundation (ASF) under one ~ or more contributor license agreements. See the NOTICE file ~ distributed with this work for additional information ~ regarding copyright ownership. The ASF licenses this file ~ to you under the Apache License, Version 2.0 (the ~ "License"); you may not use this file except in compliance ~ with the License. You may obtain a copy of the License at ~ ~ <http://www.apache.org/licenses/LICENSE-2.0> ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License.
12. downgrade to a failure
13. \* \* Override this if the filesystem is not case sensitive \* @return true if the case detection/preservation tests should run
14. allowed

**git\_commits:**

1. **summary:** HADOOP-9361: Strictly define FileSystem APIs - OpenStack portion  
**message:** HADOOP-9361: Strictly define FileSystem APIs - OpenStack portion git-svn-id: <https://svn.apache.org/repos/asf/hadoop/common/branches/branch-2.5/@1607615> 13f79535-47bb-0310-9956-ffa450edef68

**github\_issues:****github\_issues\_comments:****github\_pulls:****github\_pulls\_comments:****github\_pulls\_reviews:****jira\_issues:**

1. **summary:** Strictly define the expected behavior of filesystem APIs and write tests to verify compliance  
**description:** {{FileSystem}} and {{FileContract}} aren't tested rigorously enough -while HDFS gets tested downstream, other filesystems, such as blobstore bindings, don't. The only tests that are common are those of {{FileSystemContractTestBase}}, which HADOOP-9258 shows is incomplete. I propose # writing more tests which clarify expected behavior # testing operations in the interface being in their own JUnit4 test classes, instead of one big test suite. # Having each FS declare via a properties file what behaviors they offer, such as atomic-rename, atomic-delete, umask, immediate-consistency -test methods can downgrade to skipped test cases if a feature is missing.

**jira\_issues\_comments:**

1. These tests must all be enabled/disabled dynamically by the FS specific subclasses, so that tests that can run against a filesystem for any reason (e.g. no S3 credentials) can be downgraded to a skip -which then appears in the test reports. This will make clear that tests were not run, whereas today the JUnit3-derived tests are named to not match the \*Test pattern, and must be explicitly run by hand.
2. **body:** need to be able to generate HTML/PDF from markdown if the docs are to be done in markdown  
**label:** documentation
3. rename base JIRA
4. This is my first prototype of a contract-driven FS test suite. Every FS has to implement AbstractFSContract, which provides an FS factory, test working dir and report whether various options are supported (and eventually, limits). Options include things like {{supports-unix-permissions}} and {{is-case-sensitive}}, as well as test options like {{root-tests-enabled}} -that being a flag which, if set, enables tests to do things to a root dir like renaming and deleting it. There is a contract for Local FS, which fail because the seek operations of local don't quite follow the expectations of HADOOP-9495, which is something to consider. That FS contract dynamically chooses case sensitivity and unix-permissions features based on the OS it is running.
5. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12590695/HADOOP-9361-001.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 15 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1154 javac compiler warnings (more than the trunk's current 1153 warnings). {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1

release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common: org.apache.hadoop.fs.contract.localfs.TestLocalSeekContract {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2735//testReport/> Javac warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2735//artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2735//console> This message is automatically generated.

6. **body:** This patch # specifies the a hadoop filesystem # begins the notion of FS contract tests, with a Hadoop-compatible filesystem modeled as a set of paths mapping to data and metadata. # defines a few initial tests: directory operations, seek operations (lifted from the HADOOP-8545 tests), and a test suite for {{FileSystem.concat()}} h4. Tries to define the behaviour of the {{FileSystem}} class rigorously # It does this with a model of a filesystem as a set of paths mapping to metadata and data, which lets us use set-theory to select parts of the filesystem, and in describing how a modified filesystem is derived from its predecessor as a result of an operation. # specifying the preconditions and postconditions of the main operations -based on the behaviour of HDFS. # begins the notion of FS contract tests, with a Hadoop-compatible filesystem modeled as a set of paths mapping to data and metadata. # defines a few initial tests: directory operations, seek operations (lifted from the HADOOP-8545 tests), and a test suite for {{FileSystem.concat()}} Every filesystem to be tested must implement a subclass of {{AbstractFileSystemContract}} which contains a factory for their {{FileSystem}} instances (and could easily do the same for {{FileContext}}), as well as a method {{boolean supports(feature:string)}} that returns true/false for features like {{"supports-append"}}. For the local fs and HDFS, the definition of supported features is done from XML config files, using keys like {{fs.filesystem.contract.supports-append}}. This allows the future option for these declarations to move into the core- XML files. The {{LocalFSContract}} class updates some of its supported features ({{is-case-sensitive}}, {{supports-unix-permissions}}) based on the features of underlying FS. Every contract also has a couple of test-only features: {{supports-root-tests}} (can you do things like {{rm /}} in a test run?) and a method to see if the tests are enabled ({{boolean enabled()}}). This lets contract test suites disable themselves if they can't run (i.e when the logon details for a blobstore aren't in the test configurations). The enabled() option is checked in the test setups; the other features can be tested in the specific tests -and all downgrade to a skipped test if not set. This makes it visible which tests have not actually run. h4. Instantiates contract tests for the localFS and HDFS. Note that LocalFS fails the seek tests, as it downgrades negative seeks to a no-op. Having written these tests, I can see some limitations of the process. \* the tests have to look for the loosest exception type coming back from a failure -e.g. {{IOException}} over {{EOFException}}. \* needs a story for timeouts: we must have these to deal with blobstores &c, and either need a way to allow FS contracts to define these, let the FS specific tests define them, or just have some base timeouts large enough to deal with blobstores with operations like {{delete(dir)}} being {{O(children(dir))}} h4. Specification Syntax I'm not convinced the syntax for specifications is great, nor am I sure I'm even using it consistently. I've just had to do something minimal that fits into unformatted text in a {{.apt}} file: {code} FileSystem.listStatus(Path P, PathFilter Filter) A PathFilter is a predicate function that returns true iff the path P meets the filter's requirements. Preconditions ----- #path must exist exists(FS, P) || throw FileNotFoundException ----- Postconditions ----- isFile(FS, P) && Filter(P) => [FileStatus(FS, P)] isFile(FS, P) && !Filter(P) => [] isDir(FS, P) => [all C in children(FS, P) where Filter(C) == true] ----- {code} This blurs C/Java symbols with bits of set theory, and trying to define what exceptions to throw on failed preconditions is something that really needs improving. If we could use LaTeX we could do proper set syntax, though we'd still need a consistent language for defining filesystem implementation behaviours. {code} isFile(FS, P) \land \neg Filter(P) \rightarrow \emptyset isDir(FS, P) \rightarrow \left\{ \forall \text{all } C \in \text{children}(FS, P) : Filter(C) \right\} {code} This is actually harder to read. One other tactic could be to move the specification entirely to javadocs, with something at the package level for the core model & notation, then add precondition/postcondition specifics as preformatted areas in the docs. This would keep the spec by the signature, increase likelihood of maintenance, and for those people who understood the syntax, be able to understand what the methods do.

**label:** test

7. contract tests for local and HDFS. Local seeks wrong
8. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12591079/HADOOP-9361-002.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 22 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1154 javac compiler warnings (more than the trunk's current 1153 warnings). {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.fs.contract.localfs.TestLocalSeekContract {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2747//testReport/> Javac warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2747//artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2747//console> This message is automatically generated.
9. **body:** Thanks Steve for starting this. Personally I'm really glad to see more abstract testing-against-the-contract efforts for the file systems in Hadoop so it's great to see this. My comments upon first reading of the code (I didn't read the specs yet so no comments on those): # Personal aesthetic point: I'd have personally preferred if the contract was not in XML config, but just in code; and that isSupported() just took an enum that provided the feature. Code is much easier and simpler to read in this case, and there's no real need to "configure" a contract for a file system. And already you modify the configuration based on code in e.g. LocalFSContract, so it's just easier to read if it was all in code. # Typo: SUPPORTS\_CONTAT should be SUPPORTS\_CONCAT # In assertPathExists(), I think you meant to include the ls() content in the fail() message rather than just call it. # In testConcatOnSel() - missed failing if an error isn't thrown. # It would be great to catch() more specific exceptions in the Concat tests - i.e. have the contract specify what exception to expect as well as just an exception being thrown.

**label:** documentation

10. This is up as a work in progress; if you add the right details to the auth-keys.xml file in common/src/test/resources it will test S3n and ftp filesystems as well as local; HDFS adds its tests too. Some variances # HDFS won't let you rm / even if empty, or rm -rf /. # FTP throws FileNotFoundException if you try to delete a file that isn't there. # FTP (the back end?) will overwrite a directory with a file creation. # {{RawLocalFS.rename()}} will try to do a {{File.rename()}} operation and fall back to a copy, this is the outstanding issue from HDFS-303, "What should consistent rename actions be" # S3n will not only let you overwrite a dir with a file (side effect of how blobstores use 0-byte files as dir markers), it will do this even if the destination has children. It could check for that
11. marking as rename needs to use definition of HADOOP-6240 for its definition & tests
12. The latest patch now has tests for : create, open, delete, mkdir and seek. I'm ignoring the rename tests as I need to fully understand what HADOOP-6240 has defined first. h3. seek # I've been through the code and fixed wherever a -ve seek was either ignored or raised an {{IOException}} into an {{EOFException}}. This included changes to {{ChecksumFileSystem}}, {{RawLocalFileSystem}}, {{BufferedFSInputStream}} (which also handles a null inner stream without NPEing), {{FSInputChecker.java}} # pulled in the test from HADOOP-9307 to do many random seeks and reads; the #of seeks is configurable, so that remote blobstore tests don't take forever unless you want it to (or are running them in-cluster) # some filesystems let you seek over a closed stream. I've fixed the NPE in {{BufferedFSInputStream}}, not sure it is worth the effort of fixing this everywhere. h3. NativeS3 issues/changes changes \* {{Jets3tNativeFilesystemStore}} converts the relevant S3 error code {{"InvalidRange"}} into an EOFException \* Amazon S3 rejects a seek(0) in a zero-byte file; not fixed yet as you need to know the file length to do it up front. Maybe an EOFException on a seek could be downgraded to a no-op if the seek offset is 0. \* throws a {{FileAlreadyExistsException}} if trying to create a file over an existing one, and {{!overwrite}} \* I'm deliberately skipping the test where we expect creating a file over a dir to fail even if overwrite is true, because blobstores use 0-byte files as a pretend directory. \* It's failing a test which creates overwrites a directory which has children. This could be picked up (look for children if overwriting a 0-byte file) \* It fails a test that a newly created file exists while the write is still in progress; as the blobstores only write at the end of the file, it doesn't. this is potentially a race condition -we could create a marker file here and overwrite it on the close. h3. FTP I'll cover that in in HADOOP-9712 as its mostly bugs in a niche FS. h3. LocalFS \* throws {{FileNotFoundException}} when attempting to create a dir where the destination or a parent is a directory. This happens inside the JDK and has to be a WONTFIX, unless it is caught and wrapped. {code} testOverwriteNonEmptyDirectory(org.apache.hadoop.fs.contract.localfs.TestLocalCreateContract) Time elapsed: 38 sec <<< ERROR! java.io.FileNotFoundException: /Users/stevel/Projects/hadoop-trunk/hadoop-common-project/hadoop-common/target/test/data/testOverwriteNonEmptyDirectory (File exists) at java.io.FileOutputStream.open(Native Method) at java.io.FileOutputStream.<init>(FileOutputStream.java:194) at org.apache.hadoop.fs.RawLocalFileSystem\$LocalFSFileOutputStream.<init>(RawLocalFileSystem.java:227) at org.apache.hadoop.fs.RawLocalFileSystem\$LocalFSFileOutputStream.<init>(RawLocalFileSystem.java:223) at org.apache.hadoop.fs.RawLocalFileSystem.create(RawLocalFileSystem.java:286) at org.apache.hadoop.fs.RawLocalFileSystem.create(RawLocalFileSystem.java:273) at org.apache.hadoop.fs.ChecksumFileSystem\$ChecksumFSOutputSummer.<init>(ChecksumFileSystem.java:384) at org.apache.hadoop.fs.ChecksumFileSystem.create(ChecksumFileSystem.java:443) at org.apache.hadoop.fs.ChecksumFileSystem.create(ChecksumFileSystem.java:424) at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:888) at org.apache.hadoop.fs.FileSystem.create(FileSystem.java:869) at org.apache.hadoop.fs.contract.ContractTestUtils.writeDataset(ContractTestUtils.java:130) at org.apache.hadoop.fs.contract.AbstractCreateContractTest.testOverwriteNonEmptyDirectory(AbstractCreateContractTest.java:115) {code} # if you call {{mkdir(path-to-a-file)}} you get a 0 return code -but no exception is thrown. This is inconsistent with HDFS. {code} testNoMkdirOverFile(org.apache.hadoop.fs.contract.localfs.TestLocalDirectoryContract) Time elapsed: 46 sec <<< FAILURE! java.lang.AssertionError: mkdirs succeeded over a file: ls file:/Users/stevel/Projects/hadoop-trunk/hadoop-common-project/hadoop-common/target/test/data/testNoMkdirOverFile[00]

RawLocalFileStatus{ path=file:/Users/stevil/Projects/hadoop-trunk/hadoop-common-project/hadoop-common/target/test/data/testNoMkdirOverFile; isDirectory=false; length=1024; replication=1; blockSize=33554432; modification\_time=1373457007000; access\_time=0; owner=; group=; permission=rw-rw-rw-; isSymlink=false} at org.junit.Assert.fail(Assert.java:93) at org.apache.hadoop.fs.contract.AbstractDirectoryContractTest.testNoMkdirOverFile(AbstractDirectoryContractTest.java:68) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.junit.runners.model.FrameworkMethod\$1.runReflectiveCall(FrameworkMethod.java:45) at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:15) at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:42) at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:20) at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:28) at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:30) at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:263) at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:68) at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:47) at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:231) at org.junit.runners.ParentRunner\$1.schedule(ParentRunner.java:60) at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:229) at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:50) at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:222) at org.junit.runners.ParentRunner.run(ParentRunner.java:300) at org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:252) at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:141) at org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:112) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) {code} h3. HDFS Ambiguities \*you can't {rm /} an empty root dir, or {rm -rf /} a non-empty root dir. This may be a good design choice for safety; not consistent with the behaviours of all (tested) filesystems. I haven't tested FTP or local FS though, for obvious reasons (these tests are only run if you subclass the relevant test, \*and\* explicitly enable it) \* {{FileAlreadyExistsException}} is thrown instead of {{ParentNotDirectoryException}} when a {{mkdir}} is make with a parent file {code} ttestMkdirOverParentFile(org.apache.hadoop.fs.contract.hdfs.TestHDFSDirectoryContract) Time elapsed: 48 sec <<< ERROR! org.apache.hadoop.fs.FileAlreadyExistsException: Parent path is not a directory: /test/testMkdirOverParentFile testMkdirOverParentFile at org.apache.hadoop.hdfs.server.namenode.FSDirectory.mkdirs(FSDirectory.java:1906) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInternal(FSNamesystem.java:3182) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInt(FSNamesystem.java:3141) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:3114) at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:692) at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdirs(ClientNamenodeProtocolServerSideTranslatorPB.java:502) at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos\$ClientNamenodeProtocol\$2.callBlockingMethod(ClientNamenodeProtocolProtos.java:4808) at org.apache.hadoop.ipc.ProtobufRpcEngine\$Server\$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:605) at org.apache.hadoop.ipc.RPC\$Server.call(RPC.java:1033) at org.apache.hadoop.ipc.Server\$Handler\$1.run(Server.java:1880) at org.apache.hadoop.ipc.Server\$Handler\$1.run(Server.java:1876) at java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.Subject.doAs(Subject.java:396) at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1489) at org.apache.hadoop.ipc.Server\$Handler.run(Server.java:1874) at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:39) at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:27) at java.lang.reflect.Constructor.newInstance(Constructor.java:513) at org.apache.hadoop.ipc.RemoteException.instantiateException(RemoteException.java:106) at org.apache.hadoop.ipc.RemoteException.unwrapRemoteException(RemoteException.java:73) at org.apache.hadoop.hdfs.DFSClient.primitiveMkdir(DFSClient.java:2324) at org.apache.hadoop.hdfs.DFSClient.mkdirs(DFSClient.java:2293) at org.apache.hadoop.hdfs.DistributedFileSystem.mkdirs(DistributedFileSystem.java:568) at org.apache.hadoop.fs.FileSystem.mkdirs(FileSystem.java:1915) at org.apache.hadoop.fs.contract.AbstractDirectoryContractTest.testMkdirOverParentFile(AbstractDirectoryContractTest.java:95) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.junit.runners.model.FrameworkMethod\$1.runReflectiveCall(FrameworkMethod.java:45) at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:15) at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:42) at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:20) at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:28) at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:30) at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:263) at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:68) at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:47) at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:231) at org.junit.runners.ParentRunner\$1.schedule(ParentRunner.java:60) at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:229) at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:50) at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:222) at org.junit.runners.ParentRunner.run(ParentRunner.java:300) at org.apache.maven.surefire.junit4.JUnit4Provider.execute(JUnit4Provider.java:252) at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:141) at org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:112) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.apache.maven.surefire.util.ReflectionUtils.invokeMethodWithArray(ReflectionUtils.java:189) at org.apache.maven.surefire.booter.ProviderFactory\$ProviderProxy.invoke(ProviderFactory.java:165) at org.apache.maven.surefire.booter.ProviderFactory.invokeProvider(ProviderFactory.java:85) at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:115) at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:75) Caused by: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.fs.FileAlreadyExistsException): Parent path is not a directory: /test/testMkdirOverParentFile testMkdirOverParentFile at org.apache.hadoop.hdfs.server.namenode.FSDirectory.mkdirs(FSDirectory.java:1906) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInternal(FSNamesystem.java:3182) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInt(FSNamesystem.java:3141) at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:3114) at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:692) at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdirs(ClientNamenodeProtocolServerSideTranslatorPB.java:502) at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos\$ClientNamenodeProtocol\$2.callBlockingMethod(ClientNamenodeProtocolProtos.java:4808) at org.apache.hadoop.ipc.ProtobufRpcEngine\$Server\$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:605) at org.apache.hadoop.ipc.RPC\$Server.call(RPC.java:1033) at org.apache.hadoop.ipc.Server\$Handler\$1.run(Server.java:1880) at org.apache.hadoop.ipc.Server\$Handler\$1.run(Server.java:1876) at java.security.AccessController.doPrivileged(Native Method) at javax.security.auth.Subject.doAs(Subject.java:396) at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1489) at org.apache.hadoop.ipc.Server\$Handler.run(Server.java:1874) at org.apache.hadoop.ipc.Client.call(Client.java:1314) at org.apache.hadoop.ipc.Client.call(Client.java:1266) at org.apache.hadoop.ipc.ProtobufRpcEngine\$Invoker.invoke(ProtobufRpcEngine.java:206) at com.sun.proxy.\$Proxy16.mkdirs(Unknown Source) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:163) at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:82) at com.sun.proxy.\$Proxy16.mkdirs(Unknown Source) at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.mkdirs(ClientNamenodeProtocolTranslatorPB.java:467) at org.apache.hadoop.hdfs.DFSClient.primitiveMkdir(DFSClient.java:2322) ... 37 more {code}

13. patch which contains the tests, though the ftp and s3n tests won't run unless test filesystems are provided, only local and HDFS, which will show up some of the ambiguities
14. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12591663/HADOOP-9361-004.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 52 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1154 javac compiler warnings (more than the trunk's current 1153 warnings). {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs: org.apache.hadoop.fs.contract.mock.TestMockFSContract org.apache.hadoop.fs.contract.localfs.TestLocalRenameContract org.apache.hadoop.fs.TestLocalFileSystem org.apache.hadoop.fs.contract.localfs.TestLocalSeekContract org.apache.hadoop.fs.contract.localfs.TestLocalMkdirContract org.apache.hadoop.fs.contract.hdfs.TestHDFSContract org.apache.hadoop.fs.contract.hdfs.TestHDFSRenameContract org.apache.hadoop.fs.contract.hdfs.TestHDFSRootDirectoryContract {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2759/testReport/> Javac warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2759/artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/2759/console> This message is automatically generated.
15. **body:** This updates the tests with # SwiftFS contract (now that HADOOP-8545 is in) # appends contract test, with HDFS the only tested implementation. There's an issue with append that this test shows up (it fails), which is: what happens if a file that is being appended to is renamed? In HDFS, the answer appears to be "it keeps the old name", though the test doesn't explore in detail what has happened.  
**label:** test
16. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12605475/HADOOP-9361-005.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 65 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1536 javac compiler warnings (more than the trunk's current 1535 warnings). {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 2 new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.fs.contract.mock.TestMockFSContract org.apache.hadoop.fs.contract.localfs.TestLocalRenameContract org.apache.hadoop.fs.contract.localfs.TestLocalSeekContract org.apache.hadoop.fs.contract.localfs.TestLocalMkdirContract org.apache.hadoop.fs.contract.hdfs.TestHDFSContract org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract org.apache.hadoop.fs.contract.hdfs.TestHDFSRootDirectoryContract {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3143/testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3143/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-openstack.html> Javac warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3143/artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3143/console> This message is automatically generated.
17. Patch with spec consistent with almost all HDFS behaviour, s3 and localfs set up to throw tighter exceptions on some failures, and to throw EOFException on seek(-negative). h2. LocalFS behaviours to resolve 1. attempting to mkdir over an existing file returns false instead of raising an exception. propose: raise an exception. Nobody ever checks the return code from mkdirs after all -so we should update it to be on a par with HDFS. 2. you can seek on an stream valid after a close() options: \* fix \* ignore on the basis reads or writes will fail when attempted 3. if you rename a file over an existing file, the operation succeeds. This is what bash does. propose: document as the less preferred option; relax test to permit with a warn h2. HDFS contract test behaviours 1. if you open a stream for append, rename the file and then do the append, the old filename remains. Propose: specify the outcome as "undefined" 2. if you attempt to rename a file that doesn't exist to a path in the same directory, it returns false, rather than raising a FileNotFoundException I'm assuming here that the dest path is being checked before the source. I'd consider this an error 3. delete("", true) returns false and doesn't delete anything I've documented this as valid behaviour, and noted it is what HDFS does.
18. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12623691/HADOOP-9361-006.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 65 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1547 javac compiler warnings (more than the trunk's current 1546 warnings). {color:green}+1 javadoc{color}. The javadoc tool did not generate any warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.fs.contract.localfs.TestLocalSeekContract org.apache.hadoop.fs.contract.localfs.TestLocalMkdirContract org.apache.hadoop.fs.contract.mock.TestMockFSContract org.apache.hadoop.fs.contract.localfs.TestLocalRenameContract org.apache.hadoop.hdfs.server.namenode.ha.TestHASafeMode org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract org.apache.hadoop.fs.contract.hdfs.TestHDFSRenameContract {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3442/testReport/> Javac warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3442/artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3442/console> This message is automatically generated.
19. **body:** Updated patch # explicit option for an FS to not fail on a {{seek()}} on a closed file, only on the read that follows. # add raw local tests that bypass the {{ChecksumFilesystem}} -mostly to see where quirks lay -and a couple that go down to Java's {{File}} class just to make sure. # more tests on rename behaviour h2. Outstanding "ambiguities" h3. Major: rename file onto file On OS/X, LocalFS (and RawLocal) let you rename a source file over a destination file; the source data becomes accessible at the source path. This is what {{mv}} does at the command line, so presumably it's legitimate, even if HDFS and the blobstore APIs all reject this. This is a major difference in semantics from HDFS and Posix filesystems -code I've written to assume that renaming a file fails if the destination exists works well to implement some implicit concurrency control on HDFS, but would not work on a Posix FS. We have the option of making the rename operation stricter by explicitly adding a check into rawlocal, but there's still a race condition between any check and the OS-level rename action. What does that leave? It leaves documenting this somewhere for end-users. h3. Minor: RawLocal returns true if you attempt to delete a nonexistent path; everything else (including {{File.delete()}} returns true. This behaviour comes from {{FileUtil.delete(f)}}, which does not check for a file existence first, and when it gets false back from {{File.delete()}} generates the return code {{!File.exists()}}. That is, the semantics of fully delete are "return true if there is no directory at the end of the operation" More subtly, it there is a small a race condition where you could accidentally recursively delete a directory by attempting to delete a nonexistent file while another process is creating a directory of the same name This is because the check for {{!isFile()}} and !recursive are false when the file does not exist, but by the time the delete operation starts the rename has created a directory tree. {code} File f = pathToFile(p); if (f.isFile()) { return f.delete(); } else if (!recursive && f.isDirectory() && (FileUtil.listFiles(f).length != 0)) { throw new IOException("Directory " + f.toString() + " is not empty"); } return FileUtil.delete(f); } {code} Adding an existence check at the start of the sequence produces a consistent return code and eliminates this aspect of the race condition -this patch does exactly that. There is still a minor race: adding an entry to a directory after the empty check and before the fullyDelete call. the recursive flag logic should really be moved into a {{FileUtil}} method itself.  
**label:** code-design
20. this patch will fail rename file-on-file operations in the local and rawlocal FS, as it expects the operations to fail -but for these two they don't
21. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12636110/HADOOP-9361-007.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 75 new or modified test files. {color:red}-1 javac{color:red}. The patch appears to cause the build to fail. Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3692/console> This message is automatically generated.
22. patch fixes BufferedFSInputStream NPE on a seek on a closed file
23. Updated patch # fixes the filesystem compilation problem (some case issue not picked up on OSX) # contains a (~80% done) formal description of what FSDataInputStream implementations must do.. # fixes HADOOP-10419, "BufferedFSInputStream NPEs on getPos() on a closed stream" by raising an IOE instead
24. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12636263/HADOOP-9361-008.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 75 new or modified test files. {color:red}-1 javac{color:red}. The patch appears to cause the build to fail. Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3698/console> This message is automatically generated.

25. HADOOP-10440 shows I should add some tests to verify that position moves exactly the #of bytes requested on all read operations, and on a read where there isn't enough data, stay the same
26. -009 patch; specifies the {{FSDInputStream}}, including {{PositionedReadable}} # the base implementation of those methods don't check for negative values # although the javadocs say "thread safe", not all the implementations are
27. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12638079/HADOOP-9361-009.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 75 new or modified test files. {color:red}-1 javac{color:red}. The patch appears to cause the build to fail. Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3733/console> This message is automatically generated.
28. updated patch # still trying to work out while its not compiling remotely # use contract/auth-keys.xml for all auth keys -makes it easy to test against remote filesystems without editing test/resources/core-site.xml (and so stops you accidentally checking in keys) # includes HADOOP-10458 for swift to throw {{FileAlreadyExistsException}} # logic to detect and downgrade blobstore quirks improved
29. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12638246/HADOOP-9361-011.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 76 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1486 javac compiler warnings (more than the trunk's current 1485 warnings). {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.fs.contract.mock.TestMockFSContract org.apache.hadoop.fs.contract.rawlocal.TestRawLocalSeekContract org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract org.apache.hadoop.fs.contract.hdfs.TestHDFSRenameContract The following test timeouts occurred in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.hdfs.TestDFSClientRetries {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: [https://builds.apache.org/job/PreCommit-HADOOP-Build/3736/testReport/Javac warnings](https://builds.apache.org/job/PreCommit-HADOOP-Build/3736/testReport/Javac%20warnings): <https://builds.apache.org/job/PreCommit-HADOOP-Build/3736/artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3736/console> This message is automatically generated.
30. hi steve progress looks great. Have started some on both HADOOP-10463 and HADOOP-10461 which i think both will be able to leverage the declarative part that this adds. - What did you mean "doesn't compile remotely" ? Let me know if i can lend a hand with anything. looks like tests are failing but compilation works. - let me know if i can lend a hand anywhere :)
31. This test suite shows up HDFS-6262 -HDFS does not throw a {{FileNotFoundException}} if the source of a rename doesn't exist
32. linking as a dependent on HDFS-4258 -handling of rename during operations on open files -including append- Without this, a test of HDFS append+ rename fails: {code} ----- T E S T ----- Running  
org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract Tests run: 5, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 3.481 sec <<< FAILURE! - in org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract testRenameFileBeingAppended(org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract) Time elapsed: 0.044 sec <<< FAILURE! java.lang.AssertionError: renamed destination file does not exist: not found hdfs://localhost:54005/test/test/renamed in hdfs://localhost:54005/test/test at org.junit.Assert.fail(Assert.java:93) at org.apache.hadoop.fs.contract.ContractTestUtils.assertPathExists(ContractTestUtils.java:587) at org.apache.hadoop.fs.contract.AbstractFSContractTestBase.assertPathExists(AbstractFSContractTestBase.java:254) at org.apache.hadoop.fs.contract.AbstractAppendContractTest.testRenameFileBeingAppended(AbstractAppendContractTest.java:127) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:606) at org.junit.runners.model.FrameworkMethod\$1.runReflectiveCall(FrameworkMethod.java:45) at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:15) at org.junit.runners.model.FrameworkMethod.invokeExplosively(FrameworkMethod.java:42) at org.junit.internal.runners.statements.InvokeMethod.evaluate(InvokeMethod.java:20) at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:28) at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:30) at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:263) at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:68) at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:231) at org.junit.runners.ParentRunner\$1.schedule(ParentRunner.java:60) at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:229) at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:50) at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:222) at org.junit.internal.runners.statements.RunBefores.evaluate(RunBefores.java:28) at org.junit.internal.runners.statements.RunAfters.evaluate(RunAfters.java:30) at org.junit.runners.ParentRunner.run(ParentRunner.java:300) at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:264) at org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:153) at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:200) at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:153) at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:103) {code}
33. patch rebased to trunk; minor changes to the append test to see exactly what HDFS is up to (its up to HDFS-4258 -rename of an open file doesn't pick up new name) Pending a decision on what to do with HDFS-6262 - i.e. fix it or not - this patch should be ready to go in. It's not complete coverage of the filesystem semantics, but it can be extended over time, with the new test framework.
34. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12641004/HADOOP-9361-012.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 72 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1289 javac compiler warnings (more than the trunk's current 1288 warnings). {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.hdfs.server.balancer.TestBalancerWithNodeGroup org.apache.hadoop.fs.contract.hdfs.TestHDFSRenameContract org.apache.hadoop.fs.contract.hdfs.TestHDFSAppendContract org.apache.hadoop.hdfs.server.namenode.ha.TestDFSUpgradeWithHA {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: [https://builds.apache.org/job/PreCommit-HADOOP-Build/3818/testReport/Javac warnings](https://builds.apache.org/job/PreCommit-HADOOP-Build/3818/testReport/Javac%20warnings): <https://builds.apache.org/job/PreCommit-HADOOP-Build/3818/artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3818/console> This message is automatically generated.
35. **body:** checking for null in S3 stream close is consistent with tightening up of other streams & makes close() more robust  
**label:** code-design
36. This iteration has the following main changes # docs cover testing and extending the specification # tests have options for specific behaviours of HDFS and local fs on rename corner cases (overwrite, destination is nonexistent directory) # FTP mostly works, except for the detail that rename() fails consistently. # Swift FS exceptions are in sync with what the contract tests expect, as well as the existing tests. # S3N has all the fixes needed for HADOOP-10589 -as well as the tests to show the issue. I think this is complete enough to go in -if jenkins is happy. We just need to split up the individual patches into manageable units : core+ local ,hdfs, s3, ftp, swift for actual application. Given the state of s3, I'd give that priority. If someone wants to run these tests, have a look at the testing.md file -it explains how to do it.
37. Hey Steve, this is a big patch but I'd like to help review. Do you have any kind of feedback in particular you're looking for? I don't have exposure to the various FileSystems besides local and HDFS, but I can at least look for potential compat issues (I see return types and exception types being changed), and also proofread the documentation you added. Running the Swift and S3 tests might be a bit hard too, so I'll just trust you that they work :) If you want to attack this piece by piece, I'll also wait for the patch split.
38. Hi Andrew I can help to review it: it's an important part of the hdfs initiative. Possibly we could review it In person at Hadoop summit if you are going, as I will be there for the week.
39. I wasn't planning on attending Summit, but my quick skim of the patch was that the bulk of it was testing and documentation, which I think I can get through without too much hand holding. I'd need to read through it anyway before we could talk about FS semantics, which is I think the real meat of this patch.
40. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12647652/HADOOP-9361-013.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The

- patch appears to include 77 new or modified test files. {color:red}-1 javac{color}. The applied patch generated 1279 javac compiler warnings (more than the trunk's current 1278 warnings). {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:red}-1 findbugs{color}. The patch appears to introduce 1 new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.hdfs.server.namenode.TestStartup {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: [https://builds.apache.org/job/PreCommit-HADOOP-Build/3985/testReport/Findbugs warnings](https://builds.apache.org/job/PreCommit-HADOOP-Build/3985/testReport/Findbugs%20warnings): <https://builds.apache.org/job/PreCommit-HADOOP-Build/3985/artifact/trunk/patchprocess/newPatchFindbugsWarningshadoop-common.html> Javac warnings: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3985/artifact/trunk/patchprocess/diffJavacWarnings.txt> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3985/console> This message is automatically generated.
41. looking at the warnings. I think that test failure is a false alarm, but will resubmit to see; I've also tuned s3n's exception logic a bit -that's a bit of a trouble spot. Andrew, most of this is documentation -what HDFS does and where other filesystems differ within their "correct" operation, fixes for the other filesystems to make them more consistent with HDFS and more robust -especially to {{seek()}} and {{close()}}. For HDFS, all that changes is that out of range seeks are raised as {{EOFException}} instead of simple {{IOException}} - with the text unchanged in case people were looking for that in their code. The thing I'd like input on is actually that specification. Does it make sense -and can you see obvious mistakes in it. Rename() is the trouble spot, incidentally -in the spec, the implementation and the tests. As for the tests, they've got a more contract-driven architecture, with an XML file for each FS defining behaviour, e.g [hdfs.xml]<https://github.com/stevcloughran/hadoop-trunk/blob/stevcl/HADOOP-9361-filesystem-contract/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/contract/hdfs.xml>. These are in the test JARs to stop them being used by code, though I'd like them (somehow) to get down to any functional tests in Bigtop. Have a look at [that specification]<https://github.com/stevcloughran/hadoop-trunk/tree/stevcl/HADOOP-9361-filesystem-contract/hadoop-common-project/hadoop-common/src/site/markdown/filesystem> and call out trouble spots. I'll be in the US after hadoop summit if you want to talk on it, though JIRA is where detailed feedback should go for the historical record.
42. **body:** Patch with changes # fixed a findbug warning about unused private method # fixed a deprecation warning about use of {{Path.makeQualified()}} # reviewed the new S3N exception translation logic and enhanced it with a change in the recursion termination process and with no wrap of any final IOE at that end stage  
**label:** code-design
43. **body:** "These are in the test JARs to stop them being used by code, though I'd like them (somehow) to get down to any functional tests in Bigtop." ^^ can you clarify what you meant by that? Certainly there is an overlap of stuff bigtop should be aware of here so it will be good to update or fs tests like TestCli jiras in context of this.  
**label:** test
44. {color:green}+1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12647777/HADOOP-9361-014.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 77 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3989/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/3989/console> This message is automatically generated.
45. Jay -what I'm thinking of is that functional filesystem tests written in the bigtop project may want to make use of some of the information. What is there today is low-level, and covers details needed for the basic filesystem tests, but what would be interesting is information on the limits of the filesystem # the maximum number of files in a directory # the maximum path length in the filesystem # the maximum length of a directory entry # the maximum file size these can be used for stress tests -though of course they'd need to be overridable by individuals/test runs. this is stuff that could/should be kept together with the other details. The hadoop build keeps XML and properties resources out of the test JARs, which is normally good: it stops log4j files sneaking in, or people's site.xml files with login details accidentally being published. its just we may want -at some time in the future- to publish this -for testers rather than apps in general
46. Thanks Steve. I wasn't planning on attending Summit, but if there's a particularly good day (like say an HDFS meetup), I can try and come. I suspect that we'll be able to work most stuff out via JIRA and upstream lists though. It seems a little scary that essentially just the two of us could push through something as powerful as a FileSystem spec without sign-off from the rest of the community, but I guess at this point everyone's had ample time to comment if they care. We should probably re-ping the dev list before actual commit though just to be safe. This is also something we might consider baking in trunk before backporting to branch-2. There'll still be immediate value too, since it'll be running on precommit at that point.
47. any meet up would come after. # it's not a change, just a documentation of what HDFS does and tightening of the others -driven by tests on their behaviour # Suresh has spent a few hours reviewing it, don't worry. # we need the fixes to S3N in as soon as possible, as the jets3t migration triggers NPEs without tightened error handling. Everything else is really tests -which don't break anything- and tightened exceptions. Which, as they are made closer to HDFS, "shouldn't".
48. Hey Steve, would it be appropriate if I attached an addendum patch with doc improvements? I think that'd be easier than posting a gigantic review of "capitalize this" or "reword that". You could then re-integrate them if you agree. I'll still post any code review comments here on JIRA.
49. +1 to a patch of the docs, I'll merge them in and then look at the diff in the IDE. Code reviews welcome -as I warned, it's S3 that has the most changes, though FTP has some too, as does swift. -steve
50. **body:** Cool, thanks Steve. Here's a patch with a bunch of docs changes. I applied your base patch to git hash b7da01dea546746e34195882df0dee789bc2e3c5 (the trunk HDFS-6268 commit), then made this one on top of that. There's a lot of content here, so I won't claim to have caught everything, but hopefully it'll help you refine the text. I also had some other higher level comments about the text (haven't looked at the code yet): High-level/misc: - [Wikipedia] ([https://en.wikipedia.org/wiki/Dash#En\\_dash\\_versus\\_em\\_dash](https://en.wikipedia.org/wiki/Dash#En_dash_versus_em_dash)) is enlightening on the subject of em and en dashes, it seems like an unspaced em-dash ({&mdash;}) may be slightly more canonical than a spaced en-dash ({&nbsp;mdash;}). I used the unspaced emdash. - I assume we wanted to say "FileSystem" with camel casing in a lot of places. Please check the rest. - Also tried to use the Oxford comma everywhere, it varied in your writing. - A table of contents would be useful for the bigger pages. Introduction - Maybe should define what a "blobstore" is? Opinions vary, I've heard people call HDFS a "blobstore" before because it's append-only and not POSIX. - What is "immediate consistency"? - In Atomicity, note that we don't actually implement mkdir in FileSystem, but makedirs. - Link for "one-copy-update-semantics"? Also, while HDFS does support this, it requires some hoops with reopening the DFSInputStream to get the new file length of files being written. Bit of a caveat. - Concurrency and consistency are separate sections? I'm not sure what the difference is. - "HDFS: 8000", is this bytes or characters? - The "undefined limits" and "undefined timeouts" come off as commentary, should we be sprinkling SHOULD around, or giving more advice about "typical" expectations? Notation - In Exceptions, it says that you can provide a set of exceptions, but you used list syntax. Wasn't sure here, so I switched it to set syntax (curly braces). Model: - I think "path component" is a more standard term than "path element"? - Paths are URIs in Hadoop, is that worth mentioning here? Path URI normalization is also complicated, things like extra slashes and "...". are (I believe) sometimes normalized out. - Where is it specified how to turn a path into path components? There's also a need to strip out parts of the URI like the scheme and authority. - ancestors doesn't have preconditions - I'm not sure we'll ever get symlinks in FileSystem to be honest, I'd consider just removing these references. - isDescendant definition refers to itself, is this right? - Dunno how to parse the "File references statement" about data dictionary - dangling sentence in "User home" Filesystem: - Looking at the code, getBlockLocations takes a Path, and throws a FileNotFoundException if it doesn't exist. - The cluster topology stuff seems like a non-sequiter, maybe more explanation? - append Postconditions, just "FS"? - "the file and its data is still", dunno how to parse this - Worth mentioning permissions as related to recursive delete? I know permissions are assumed, but IIRC the atomicity still holds. - I'm wondering about the use of MUST with regard to atomicity of recursive delete. In other places, you mention that behavior is undefined or implementation-dependent, but here you say it's a MUST but these other FileSystems don't support it. FSDIS: - Close in a distributed filesystem is a thorny problem. Have you seen HDFS-4504? I've also heard of Flume and HBase errors related to close continually throwing IOException, not sure of the current status. - Mention and formatting of exceptions is not uniform. InputStream.read is one example, it doesn't use the list syntax, and NullPointerException is mentioned in the preconditions box but not the exceptions box. - Invariants are not uniformly formatted - Formatting of true and false is not uniform (e.g. {{True}} and true in text). - seekToNewSource, irregular use of terms "files" and "blocks", not sure if you wanted to avoid talking about blocks entirely, or instead wanted to define both terms - I think we could use some more rigor in the Consistency section, some of it seems underspecified Testing: - Note that "LocalFS" is a FileContext class, LocalFileSystem is probably what you want. Haven't looked at the code yet, but might need to rename some things. - Rather than saying "Windows and OS/X filesystem", can we go further and say, e.g. HPFS, NTFS, FAT, etc? - I can't parse the paragraph beginning with "A recommended strategy" - Definition of concurrency vs consistency again?  
**label:** code-design
51. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12648399/HADOOP-9361-awang-addendum.patch> against trunk revision . {color:red}-1 patch{color}. The patch command could not apply the patch. Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4009/console> This message is automatically generated.

52. Should HADOOP-9361 HADOOP-9371 now be linked ? -- which is the blocker?
53. This is revision -015 of the patch # incorporates all of Andrew's modifications. Andrew - thanks for putting in the effort! # added a section on object stores in the introduction, to clarify how they are different. Once we add a 'Blobstore' marker to object stores, we can expand that a bit more.
54. {color:green}+1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12648875/HADOOP-9361-015.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 77 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4024/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4024/console> This message is automatically generated.
55. **body:** Hi Steve, I'm going to take it on faith that you applied all the doc improvements, since honestly there's too much to look through :) Going to focus on code questions this time around. It's still hard for me to run the S3 / Swift tests since I'd have to get credentials, but I can do it if needed. \* Only big feedback I have, is there some way we could have a single parent test for each FS that runs all its contracts? All the concrete TestFooBarContract classes are basically copy paste, I'd like to get rid of them. It's also harder for poor me to run all the HDFS contract tests. \* site.xml, link is still not quite right, should be filesystem/index.html \* FSExceptionMessages, should we re-use the Java messages for these exceptions? I think they're a bit different. There are also a bunch of places where these strings \*aren't\* used, but unifying it everywhere is a tall order. Maybe a follow-on JIRA. Another maybe nice follow-on would be having a factory method for things like FileAlreadyExistsException where we stitch something to a string. \* I'd prefer to flip "fs.contract.supports-seek-past-eof" so we can have all trues for HDFS's contract. Strawman, "supports-exception-on-seek-past-eof"? \* Jets3native: typo: maxListingLengthm \* Jets3native: there are two handleBlahException methods that just delegate to handleException. Can we just get rid of them? \* NativeS3, read's LOG.info has an extra "{}" \* Some Contract class javadocs are inconsistent. HDFSContract's class javadoc says "Local filesystem". SwiftContract mentions S3N. I think we're pretty close, modulo the big one. Thanks again for the effort here.  
**label:** documentation
56. FYI, I've opened HADOOP-10461, which aims to wrap tests using Junit "Enclosed" annotations. Would that (as a follow up JIRA) Be an effective fix to your issue [~andrew.wang]
57. **body:** Hey Jay, perhaps, but it looks like there'd still be the boilerplate for each inner class. Ideally, all we'd for each FS would be the xml file and a small parent test that provides the filesystem to be tested. Let's see how close we can get to that.  
**label:** test
58. Shall we nevertheless fold your needs into HADOOP-10461? Or do you think it's a blocker?
59. **body:** Let's give Steve a little while to rework the patch first :) I'm not hung up on this point, but I'd like to see this attempted pre-commit because it might result in a lot of code churn. Let's also just say that follow-on JIRAs are not always finished ;)  
**label:** code-design
60. h3. Validating S3/Swift tests. {quote} It's still hard for me to run the S3 / Swift tests since I'd have to get credentials, {quote} These tests cost ~\$0 to run as they don't persist data; you can use your own private logins. I'd recommend >1 swift provider (I use RAX US, RAX UK and HP Cloud for their different auth, consistency and throttling). S3 has different consistency models for US-east (worst) and everywhere else. You need to create buckets in different places there. h3. test groupings {quote} Only big feedback I have, is there some way we could have a single parent test for each FS that runs all its contracts? All the concrete TestFooBarContract classes are basically copy paste, I'd like to get rid of them. It's also harder for poor me to run all the HDFS contract tests {quote} -1 to any unified supertest, as it doesn't isolate things the way having tests per operation does. You get to implement the features you want, and don't do a test for a RootContract if you don't want your root FS deleted, Append if you don't do append. etc. You also get to debug something {{TestHDFSRenameContract}} # We can do well known names like {{TestHDFSContract}}, I can see that I've already done this for the {{TestNativeS3\*}} tests. # There's Junit Categories[ <http://junit.org/javadoc/4.11/org/junit/experimental/categories/Categories.html>]. Maven [does now support this](<http://stackoverflow.com/questions/3100924/how-to-run-junit-tests-by-category-in-maven>) FWIW, categorizing the Hadoop tests would be a good idea at some point in the future anyway "fast, slow, scale" # There's JUnit suites -but use them and you will end up with your tests running if the suite test runs -and again if run individually. I'd go for the naming, if everyone is happy with that and come up with a good name for the HDFS tests that don't class with other things. How about {{TestHDFSContract\*}}? h3. FSExceptionMessages I picked them up from one file (HDFS?) and used there. h3. Everything else I'll do those... How about {{'rejects-seek-past-eof'}}?
61. that was a bit oddly formatted, text editor wraps vs JIRA ones. I was trying to say in having separate tests/feature is that you can work on a feature at a time, rather than run the subclass of FileSystemContractBaseTest you have today, which take long enough to make a decent coffee -or, in the case of object stores, walk to a nearby cafe and back-
62. **body:** Okay, I can see the value in that. Maybe there's some creative way to de-dupe some of the repeated @BeforeClass and @AfterClass type stuff, but it seems hard given the lack of multiple inheritance. I like your naming proposals (tests and the xml). I'll try to fire up some of these different cloud offerings in the meantime. I guess it'd be good for at least one other person to go through the exercise of running them.  
**label:** code-design
63. add code changes recommended by andrew # flag is rejects-seek-past-eof, negated and defaults to true # tests renamed to {{Test\${FS}Contract\${Operation}}}, e.g {{TestHDFSContractAppend}}. # also: tagged openstack swift as returning false on missing source file on a rename(), as it catches the FNFE and downgrades it for consistency with HDFS.
64. {color:red}-1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12650971/HADOOP-9361-016.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 77 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:red}-1 core tests{color}. The patch failed these unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack: org.apache.hadoop.ha.TestZKFailoverControllerStress {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4094/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4094/console> This message is automatically generated.
65. the ZK test is unrelated -but the fact that it is failing intermittently on Jenkins is a sign that the test has issues
66. Hi, Steve. The ZK test failure is tracked in HADOOP-10668, so no need to worry about it here.
67. seems ChecksumFileSystem.java missing the following import import java.io.EOFException;
68. Juan ... if the patch left it out (and as it compiled/ran) then the EOFException import was there at the time the patch was created. Are you trying to apply to trunk or branch-2?
69. I applied it to trunk.
70. **body:** I just tested this against S3 with s3n. all tests passed. One trivial thing is one file is left after tests are done. testRmNonEmptyRootDirNonRecursive I understand the purpose of this test is to verify the directory cannot be deleted if it's not empty. But it would be nice to have some cleanup function to cleanup all files/dirs created during tests since we use the real cloud service. In testing.md, the config file path should be "hadoop-common-project/hadoop-common/src/test/resources/contract-test-options.xml", not "hadoop-common-project/hadoop-common/src/test/contract-test-options.xml". I'll try to test with swift as well. should I use Block storage or Object storage? or it doesn't matter?  
**label:** code-design
71. cancelling patch, HADOOP-10674 replaced the java.io.\* import with an explicit import of the sole classes needed, so yes, {{EOFException}} is now missing
72. Revision 017 # fixed import of {{EOFException}} in {{ChecksumFileSystem}} -HADOOP-10674 had dropped the {{java.io.\*}} import that was picking it up # Improved s3n error reporting with (bucket, key) paths in error text wherever possible # s3n HTTP: 403 is turned into a {{hadoop.security.AccessControlException}}
73. {color:green}+1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12653489/HADOOP-9361-017.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 77 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack. {color:green}+1 contrib tests{color}. The patch passed contrib unit



tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4195/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4195/console> This message is automatically generated.

74. **body:** bq. One trivial thing is one file is left after tests are done. testRmNonEmptyRootDirNonRecursive I understand the purpose of this test is to verify the directory cannot be deleted if it's not empty. But it would be nice to have some cleanup function to cleanup all files/dirs created during tests since we use the real cloud service. will fix bq. In testing.md, the config file path should be "hadoop-common-project/hadoop-common/src/test/resources/contract-test-options.xml", not "hadoop-common-project/hadoop-common/src/test/contract-test-options.xml". thanks for spotting bq. I'll try to test with swift as well. should I use Block storage or Object storage? or it doesn't matter? object storage  
**label:** code-design
75. **body:** patch #018: \* fixed erroneous paths in testing.md \* clean up after {{testRmNonEmptyRootDirNonRecursive}}  
**label:** code-design
76. {color:green}+1 overall{color}. Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12653552/HADOOP-9361-018.patch> against trunk revision . {color:green}+1 @author{color}. The patch does not contain any @author tags. {color:green}+1 tests included{color}. The patch appears to include 77 new or modified test files. {color:green}+1 javac{color}. The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc{color}. There were no new javadoc warning messages. {color:green}+1 eclipse:eclipse{color}. The patch built with eclipse:eclipse. {color:green}+1 findbugs{color}. The patch does not introduce any new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit{color}. The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests{color}. The patch passed unit tests in hadoop-common-project/hadoop-common hadoop-hdfs-project/hadoop-hdfs hadoop-tools/hadoop-openstack. {color:green}+1 contrib tests{color}. The patch passed contrib unit tests. Test results: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4198/testReport/> Console output: <https://builds.apache.org/job/PreCommit-HADOOP-Build/4198/console> This message is automatically generated.
77. **body:** Tested two swift provider, Rackspace and HP Cloud. everything works fine. I am a little bit confused by the openstack config file. I thought with this new contract-test-options.xml, I don't need other configuration. Turn out I still need to create the auth-keys.xml and put swift service name there. I re-tested latest patch with S3N and Swift, test files are cleaned up. All looks good, thanks Steve.  
**label:** test
78. Thanks Juan for doing manual testing. I've reviewed everything in an earlier rev, and it seems like the newer revs made only minor changes. +1, thanks Steve, nice work here.
79. I have a minor question - finally had time to look at the patch again. What is the policy on using java.io exceptions with Fs. exception messages (i.e. as in {{STREAM\_IS\_CLOSED}}) versus more specific ones?
80. **body:** Andrew: thanks, will merge in today Juan: thanks for the testing. The entire Swift test suite is skipped if there's no auth-keys file -though we could migrate that to the contract-test-options.xml file. The reason for that policy is that # some of the junit 3 test suites that are subclassed for hadoop-common-test aren't skippable (junit 3, see) -this is why in Hadoop common the s3 & ftp tests don't start with Test\*. While the contract tests are designed to be self-skipping -and so logged in test reports, I left the junit 3 stuff with a Test profile -you can't really test the swift client without the settings, except for some minor unit tests Jay: tighter exceptions provide more information to clients, and lets you explicitly catch by type in your code, e.g. {{catch(EOFException e)}}. general IOExceptions with text have to be caught as IOE and then tested -and are incredibly brittle to changes in text. That's why I didn't rename text messages from exceptions in the common filesystems, even when I tightened their class: we don't know what callers are searching for the text. Whenever you can, use explicit types. I also recommend using constants for text, constants that tests can look for -and in those tests use {{Exception.toString().contains()}} as the check -not equality, so that if more details are added the test still works.  
**label:** code-design
81. SUCCESS: Integrated in Hadoop-trunk-Commit #5818 (See [<https://builds.apache.org/job/Hadoop-trunk-Commit/5818/>]) HADOOP-9361: Strictly define FileSystem APIs - HDFS portion (stevel: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607597>) \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/HDFSContract.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractAppend.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractConcat.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractCreate.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractDelete.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractMkdir.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractOpen.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractRename.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractRootDirectory.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractSeek.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/contract \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/contract/hdfs.xml HADOOP-9361: Strictly define FileSystem APIs (stevel: <http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607596>) \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/BufferedFSInputStream.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/ChecksumFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSDDataOutputStream.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSExceptionMessages.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSInputChecker.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/RawLocalFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/ftp/FTPFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/ftp/FTPInputStream.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/s3/S3FileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/s3native/Jets3tNativeFileSystemStore.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/s3native/NativeS3FileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem \* /hadoop/common/trunk/hadoop-common-project/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/extending.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/filesystem.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/fsdatainputstream.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/index.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/introduction.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/model.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/notation.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/testing.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/TestLocalFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractBondedFSContract.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractAppendTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractConcatTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractCreateTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractDeleteTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractMkdirTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractOpenTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractRenameTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractRootDirectoryTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractSeekTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractFSContract.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractFSContractTestBase.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/ContractOptions.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/ContractTestUtils.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-





[illegible]

project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/NativeS3Contract.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/ContractCreate.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/TestS3NContractDelete.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/TestS3NContractMkdir.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/TestS3NContractOpen.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/TestS3NContractRename.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/TestS3NContractRootDir.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/s3n/TestS3NContractSeek.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/resources/contract \* /hadoop/common/src/test/resources/contract/ftp.xml \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/resources/contract/rawlocal.xml \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/resources/contract/s3n.xml

85. FAILURE: Integrated in Hadoop-Yarn-trunk #603 (See [https://builds.apache.org/job/Hadoop-Yarn-trunk/603/]) HADOOP-9361: changes.txt (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607620) \* /hadoop/common/trunk \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/CHANGES.txt HADOOP-9361: site and gitignore (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607601) \* /hadoop/common/trunk/.gitignore \* /hadoop/common/trunk/hadoop-project/src/site/site.xml HADOOP-9361: Strictly define FileSystem APIs - OpenStack portion (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607600) \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/exceptions/SwiftPathExistsException.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/exceptions/SwiftNotDirectoryException.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/snative/StrictBufferedFSInputStream.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeFileSystem.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeFileSystemStore.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeInputStream.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeOutputStream.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/TestSwiftFileSystemBasicOps.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/TestSwiftFileSystemContract.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/TestSwiftFileSystemRename.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/SwiftContract.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractCreate.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractDelete.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractMkdir.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractOpen.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractRename.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractRootDir.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractSeek.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/java/org/apache/hadoop/fs/swift/hdfs2/TestV2LsOperations.java \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/resources/contract \* /hadoop/common/trunk/hadoop-tools/hadoop-openstack/src/test/resources/contract/swift.xml HADOOP-9361: Strictly define FileSystem APIs - HDFS portion (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607597) \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/HDFSContract.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractAppend.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractConcat.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractCreate.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractDelete.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractMkdir.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractOpen.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractRename.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractRootDirectory.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractSeek.java \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/contract \* /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/contract/hdfs.xml HADOOP-9361: Strictly define FileSystem APIs (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607596) \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/BufferedFSInputStream.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/ChecksumFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSDDataOutputStream.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSExceptionMessages.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/FSInputChecker.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/RawLocalFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/ftp/FTPFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/ftp/FTPInputStream.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/s3/S3FileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/main/java/org/apache/hadoop/fs/s3native/Jets3tNativeFileSystemStore.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/extending.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/filesystem.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/fsdatainputstream.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/index.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/introduction.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/model.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/notation.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/site/markdown/filesystem/testing.md \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/TestLocalFileSystem.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractBondedFSContract.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractAppendTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractConcatTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractCreateTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractDeleteTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractMkdirTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractOpenTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractRenameTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractRootDirectoryTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractContractSeekTest.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-common/src/test/java/org/apache/hadoop/fs/contract/AbstractFSContract.java \* /hadoop/common/trunk/hadoop-common-project/hadoop-

```

060. FAILURE: Integrated in Hadoop-Hdfs-trunk #1794 (See [https://builds.apache.org/job/Hadoop-Hdfs-trunk/1794/]) HADOOP-9361: changes.txt (stevel:
http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607620) * /hadoop/common/trunk * /hadoop/common/trunk/hadoop-common-
project/hadoop-common/CHANGES.txt HADOOP-9361: site and gitignore (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-
SVN&view=rev&rev=1607601) * /hadoop/common/trunk/.gitignore * /hadoop/common/trunk/hadoop-project/src/site/site.xml HADOOP-9361: Strictly define
FileSystem APIs - OpenStack portion (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607600) * /hadoop/common/trunk/hadoop-
tools/hadoop-opensource/src/main/java/org/apache/hadoop/fs/swift/exceptions/SwiftNotDirectoryException.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/main/java/org/apache/hadoop/fs/swift/exceptions/SwiftPathExistsException.java HADOOP-9361: Strictly define FileSystem APIs - OpenStack
portion (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607599) * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/main/java/org/apache/hadoop/fs/swift/snative/StrictBufferedFSInputStream.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeFileSystem.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeFileSystemStore.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeInputStream.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/main/java/org/apache/hadoop/fs/swift/snative/SwiftNativeOutputStream.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/TestSwiftFileSystemBasicOps.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/TestSwiftFileSystemContract.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/TestSwiftFileSystemRename.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/SwiftContract.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractCreate.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractDelete.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractMkdir.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractOpen.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractRename.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractRootDir.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/contract/TestSwiftContractSeek.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/java/org/apache/hadoop/fs/swift/hdfs2/TestV2LsOperations.java * /hadoop/common/trunk/hadoop-tools/hadoop-
opensource/src/test/resources/contract * /hadoop/common/trunk/hadoop-tools/hadoop-opensource/src/test/resources/contract/swift.xml HADOOP-9361: Strictly
define FileSystem APIs - HDFS portion (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607597) *
/hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/DFSInputStream.java * /hadoop/common/trunk/hadoop-hdfs-
project/hadoop-hdfs/src/test/java/org/apache/hadoop/fs/contract * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/HDFSContract.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractAppend.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractConcat.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractCreate.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractDelete.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractMkdir.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractOpen.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractRename.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractRootDirectory.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/java/org/apache/hadoop/fs/contract/hdfs/TestHDFSContractSeek.java * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-
hdfs/src/test/resources/contract * /hadoop/common/trunk/hadoop-hdfs-project/hadoop-hdfs/src/test/resources/contract/hdfs.xml HADOOP-9361: Strictly define
FileSystem APIs (stevel: http://svn.apache.org/viewcvcs.cgi/?root=Apache-SVN&view=rev&rev=1607596) * /hadoop/common/trunk/hadoop-common-
project/hadoop-common/src/main/java/org/apache/hadoop/fs/BufferedFSInputStream.java * /hadoop/common/trunk/hadoop-common-project/hadoop-

```

[illegible]