

Unit Test on Windows

Summary

When running unit tests on Windows, failures caused by file operation errors often happen. This is caused for Windows disallows deleting and overwriting a file when a process holds a handle of the file. This page describes how to avoid those problems.

Contents

- [Summary](#)
- [Contents](#)
- [hudson.util.IOException2: Failed to clean up temp dirs](#)
 - [Unable to delete slave-slaveX.log](#)
 - [other files](#)
- [503 Service Unavailable from Jenkins](#)
 - [requested operation can't be performed on file with user-mapped section open](#)
- [temporary directory grows up](#)
 - [%TMP%\hudsonXXXXXXXXXXtest is left even after tests](#)
 - [%TMP%\hudsonXXXXXXXXXXtmp is left even after tests](#)
 - [Why %TMP%\hudsonXXXXXXXXXXtmp is left?](#)
 - [No deletion code](#)
 - [Plugins are not released](#)
 - [Jenkins core caches and keep file handles](#)
 - [%TMP%\hudson-remotingXXXXXXXXXX, %TMP%\jenkins-remotingXXXXXXXXXX, is left even after tests](#)
- [Tests using symbolic links.](#)

hudson.util.IOException2: Failed to clean up temp dirs

This happens in `tearDown` phase of tests. This is often caused for JVM holds a handle of a file in a temporary directory, and Windows fails to delete the temporary directory. You can see what file failed to be deleted in `target/surefire-reports/(Test Class).txt` as following:

```
-----
Test set: hudson.plugins.parameterizedtrigger.test.NodeParametersTest
-----
Tests run: 3, Failures: 0, Errors: 3, Skipped: 0, Time elapsed: 43.091 sec <<< FAILURE!
test(hudson.plugins.parameterizedtrigger.test.NodeParametersTest) Time elapsed: 19.291 sec <<< ERROR!
hudson.util.IOException2: Failed to clean up temp dirs
    at org.jvnet.hudson.test.TemporaryDirectoryAllocator.dispose(TemporaryDirectoryAllocator.java:87)
    at org.jvnet.hudson.test.TestEnvironment.dispose(TestEnvironment.java:53)
    at org.jvnet.hudson.test.HudsonTestCase.tearDown(HudsonTestCase.java:352)
...
    at org.apache.maven.surefire.booter.SurefireStarter.runSuitesInProcess(SurefireStarter.java:91)
    at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:69)
Caused by: java.io.IOException: Unable to delete (The name of file failed to be deleted)
    at hudson.Util.deleteFile(Util.java:266)
    at hudson.Util.deleteRecursive(Util.java:316)
    at hudson.Util.deleteContentsRecursive(Util.java:227)
```

Unable to delete slave-slaveX.log

This happens when:

- Using `HudsonTestCase` and Jenkins < 1.441 or Jenkins >= 1.482 and Jenkins < 1.520 as a target of the plugin.
 - This is resolved with [24e99923e3](#)
 - This is degraded with [5dd905f2a2](#) and [4557a436bf](#)
- Using `JenkinsRule` and Jenkins < 1.520 as a target of the plugin.
 - But this does not cause test failures, just warned.
- Started a slave in the test.

This seems happens for a slave node stays opening a log file, it prevents deleting the temporary directory. This can be resolved with any of following ways:

- Use Jenkins >= 1.520.
- Add following code to your test class derived from `HudsonTestCase`.

```

private void purgeSlaves() {
    List<Computer> disconnectingComputers = new ArrayList<Computer>();
    List<VirtualChannel> closingChannels = new ArrayList<VirtualChannel>();
    for (Computer computer: jenkins.getComputers()) {
        if (!(computer instanceof SlaveComputer)) {
            continue;
        }
        // disconnect slaves.
        // retrieve the channel before disconnecting.
        // even a computer gets offline, channel delays to close.
        if (!computer.isOffline()) {
            VirtualChannel ch = computer.getChannel();
            computer.disconnect(null);
            disconnectingComputers.add(computer);
            closingChannels.add(ch);
        }
    }

    try {
        // Wait for all computers disconnected and all channels closed.
        for (Computer computer: disconnectingComputers) {
            computer.waitForOffline();
        }
        for (VirtualChannel ch: closingChannels) {
            ch.join();
        }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

@Override
protected void tearDown() throws Exception {
    if(Functions.isWindows()) {
        purgeSlaves();
    }
    super.tearDown();
}

```

- This closes log files of slaves before deleting the temporary directory.
- Same to JenkinsRule.

```

@Rule
public JenkinsRule j = new JenkinsRule() {
    private void purgeSlaves() {
        // same to above...
    }

    @Override
    protected void after() {
        if(Functions.isWindows()) {
            purgeSlaves();
        }
        super.after();
    }
};

```

other files

- Maybe you failed to close files in your plugin or in your test.
- When you do not close a file explicitly, it will be closed when the file object is destroyed in garbage collections.
- It can be resolved by calling `System.gc()` like this:

```
public void testSomething()
{
    { // this brace is needed to make all objects created in this test destroyed with GC.
      // do your test
    }

    System.gc();
}
```

503 Service Unavailable from Jenkins

This is often caused for a failure in launching Jenkins.

For the problem happens not in a thread a test function runs but in a thread launching a server, you cannot see the root cause from a console output. You can see the root cause in target/sunfire-reports/TEST-(Test Class).xml as following:

```
INFO: NO JSP Support for , did not find org.apache.jasper.servlet.JspServlet
May 18, 2013 10:58:54 AM org.mortbay.log.Slf4jLog info
INFO: Copy C:\Users\user\workspace\reproduce-jenkins-17774\target\jenkins-for-test to
work\Jetty_0_0_0_0_0_jenkins.for.test____.bj8wp3\webapp
May 18, 2013 10:58:55 AM org.mortbay.log.Slf4jLog warn
WARNING: Failed startup of context org.mortbay.jetty.webapp.WebAppContext@423e36e6{,C:
\Users\user\Desktop\workspace\reproduce-jenkins-17774\.\target\jenkins-for-test}
java.io.FileNotFoundException: work\Jetty_0_0_0_0_0_jenkins.for.test____.bj8wp3\webapp\images\48x48\computer-x.
png (requested operation can't be performed on file with user-mapped section open)
    at java.io.FileOutputStream.open(Native Method)
    at java.io.FileOutputStream.<init>(FileOutputStream.java:194)
    at java.io.FileOutputStream.<init>(FileOutputStream.java:145)
```

requested operation can't be performed on file with user-mapped section open

This happens when:

- Using Jenkins > 1.457, and Jenkins < 1.517 as a target of the plugin.
 - [7ae303f881](#) triggers this problem.
- Your test is junit3 based (using HudsonTestCase), not junit4 based (using JenkinsRule).

Jenkins webapp directory gets to be overwritten for each test method in Jenkins 1.457 in Windows. It sometimes causes an overwriting failure when some process holds a file in webapp directory. It often happens when TortoiseXXX (like TortoiseGit) is installed.

You can resolve this problem with any of following ways:

- use Jenkins >= 1.517
- switch from junit3 based tests (using HudsonTestCase) to junit4 based tests (using JenkinsRule)
 - See [Unit Test JUnit4](#) for details instructions using JenkinsRule.
- Copy HudsonTestCase.createWebServer() to your test class, and modify as following:

```

/**
 * Prepares a webapp hosting environment to get {@link ServletContext} implementation
 * that we need for testing.
 */
protected ServletContext createWebServer() throws Exception {
    server = new Server();

    WebApplicationContext context = new WebApplicationContext(WarExploder.getExplodedDir().getPath(), contextPath);
    context.setClassLoader(getClass().getClassLoader());
    context.setConfigurations(new Configuration[]{new WebXmlConfiguration(), new
NoListenerConfiguration()});
    server.setHandler(context);
    context.setMimeTypes(MIME_TYPES);
-    if(Functions.isWindows()) {
-        // this is only needed on Windows because of the file
-        // locking issue as described in JENKINS-12647
-        context.setCopyWebDir(true);
-    }
}

```

- This is a code in Jenkins 1.458. You should copy a code from the version your plugin targets.
- This reverts [7ae303f881](#) (JENKINS-12647).
 - This commit does nothing good.

temporary directory grows up

JVM uses %TMP% (java.io.tmpdir) as a temporary directory. For example, C:\Users\user\AppData\Local\Temp in Windows 8. When running a test of a plugin, following directories are created in the temporary directory, where XXXXXXXXXXXs are random numbers.

- hudsonXXXXXXXXXXtest
- hudsonXXXXXXXXXXtmp

It often happens that these directories are left even after a test finishes. There are several reasons:

- Old versions of Jenkins testing library do not delete these directories.
 - These problems are fixed in newer versions.
- These directories fail to be deleted. This happens only in Windows.

%TMP%\hudsonXXXXXXXXXXtest is left even after tests

This happens when:

- Test classes are written with JenkinsRule (junit4 based tests).
- The target is Jenkins < 1.482.
 - This is resolved with [5dd905f2a2](#)

This can be resolved with any of following ways:

- Upgrade target Jenkins to 1.482 or later.
- Override JenkinsRule#after and call TestEnvironment#dispose:

```

public class SomeTest
{
    ...

    @Rule
    public JenkinsRule j = new JenkinsRule() {
        protected void after() {
            super.after();
            if(TestEnvironment.get() != null)
            {
                try
                {
                    TestEnvironment.get().dispose();
                }
                catch(Exception e)
                {
                    e.printStackTrace();
                }
            }
        }
    };
    ...
}

```

%TMP%\hudsonXXXXXXXXXXtmp is left even after tests

Why %TMP%\hudsonXXXXXXXXXXtmp is left?

This is caused for [TestPluginManager](#) expands plugins to that directory.
There are several reasons why this directory is left:

Reason	Jenkins <= 1.509	1.510 <= Jenkins <= 1.518	Jenkins 1.519	1.520 <= Jenkins
No deletion code	x			
Plugins are not released	x	x	x	
Jenkins core caches and keep file handles			x	x

No deletion code

The deletion code is introduced in [a4d4305124](#) (1.510).
Do as [#Plugins are not released](#).

Plugins are not released

Jenkins tries to delete the directory, but you see following logs (the name of the jar file may differ):

```
...
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 30.876 sec
Failed to load native POSIX impl; falling back on Java impl. Unsupported OS.
java.io.IOException: Unable to delete C:
\Users\user\AppData\Local\Temp\hudson4413809435965306193tmp\credentials\WEB-INF\lib\findbugs-annotations-1.3.9-
1.jar
    at hudson.Util.deleteFile(Util.java:256)
    at hudson.Util.deleteRecursive(Util.java:308)
    at hudson.Util.deleteContentsRecursive(Util.java:205)
    at hudson.Util.deleteRecursive(Util.java:299)
    at hudson.Util.deleteContentsRecursive(Util.java:205)
    at hudson.Util.deleteRecursive(Util.java:299)
    at hudson.Util.deleteContentsRecursive(Util.java:205)
    at hudson.Util.deleteRecursive(Util.java:299)
    at hudson.Util.deleteContentsRecursive(Util.java:205)
    at hudson.Util.deleteRecursive(Util.java:299)
    at org.jvnet.hudson.test.TestPluginManager$1.run(TestPluginManager.java:130)

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

This directory contains jar files, JVM holds the handle of these files, and the deletion of this directory triggered by JVM fails.

You can resolve this problem with following steps:

- Trigger the release of plugins and the deletion of the temporary directory at the shutdown of JVM.

Here are sample codes:

- A utility class to delete the temporary directory at the shutdown of JVM.

```

import java.io.IOException;

import hudson.Util;

import org.jvnet.hudson.test.TestPluginManager;

/**
 * Cleanup the temporary directory created by org.jvnet.hudson.test.TestPluginManager.
 * Needed for Jenkins < 1.510
 *
 * Call TestPluginManagerCleanup.registerCleanup() at least once from anywhere.
 */
public class TestPluginManagerCleanup {
    private static Thread deleteThread = null;

    public static synchronized void registerCleanup() {
        if(deleteThread != null) {
            return;
        }
        deleteThread = new Thread("HOTFIX: cleanup " + TestPluginManager.INSTANCE.rootDir) {
            @Override public void run() {
                if(TestPluginManager.INSTANCE != null
                    && TestPluginManager.INSTANCE.rootDir != null
                    && TestPluginManager.INSTANCE.rootDir.exists()) {
                    // Work as PluginManager#stop
                    for(PluginWrapper p: TestPluginManager.INSTANCE.getPlugins())
                    {
                        p.stop();
                        p.releaseClassLoader();
                    }
                    TestPluginManager.INSTANCE.getPlugins().clear();
                    System.gc();
                    try {
                        Util.deleteRecursive(TestPluginManager.INSTANCE.rootDir);
                    } catch (IOException x) {
                        x.printStackTrace();
                    }
                }
            }
        };
        Runtime.getRuntime().addShutdownHook(deleteThread);
    }
}

```

- This may not work for some environments. In that case, following pages may be helpful:
 - <http://management-platform.blogspot.jp/2009/01/classloaders-keeping-jar-files-open.html>
 - <http://abondar-howto.blogspot.jp/2010/06/howto-unload-jar-files-loaded-by.html>

- A test class based on HudsonTestCase (junit3 based)

```

public class SomeTest extends HudsonTestCase {
    static {
        TestPluginManagerCleanup.registerCleanup();
    }
    ...
}

```

- A test class based on JenkinsRule (junit4 based)

```

public class SomeTest {
    static {
        TestPluginManagerCleanup.registerCleanup();
    }
    ...
}

```

Jenkins core caches and keep file handles

Jenkins packs files of each plugin into classes.jar to improve performance since 1.519 ([f7330d7](#)). This is caused as JVM (to be exact, URLConnection) keeps file handles of jar files even after Jenkins released them. You will see outputs like this:

```
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 62.997 sec - in hudson.plugins.copyartifact.
BuildSelectorParameterTest
java.nio.file.FileSystemException: C:
\Users\yasuke\AppData\Local\Temp\hudson7190248855254390892tmp\credentials\WEB-INF\lib\classes.jar: The process
cannot access the file because it is being used by another process.

    at sun.nio.fs.WindowsException.translateToIOException(WindowsException.java:86)
    at sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:97)
    at sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:102)
    at sun.nio.fs.WindowsFileSystemProvider.implDelete(WindowsFileSystemProvider.java:269)
    at sun.nio.fs.AbstractFileSystemProvider.delete(AbstractFileSystemProvider.java:103)
    at java.nio.file.Files.delete(Files.java:1077)
    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 hudson.Util.deleteFile(Util.java:247)
    at hudson.Util.deleteRecursive(Util.java:310)
    at hudson.Util.deleteContentsRecursive(Util.java:212)
    at hudson.Util.deleteRecursive(Util.java:301)
    at hudson.Util.deleteContentsRecursive(Util.java:212)
    at hudson.Util.deleteRecursive(Util.java:301)
    at hudson.Util.deleteContentsRecursive(Util.java:212)
    at hudson.Util.deleteRecursive(Util.java:301)
    at hudson.Util.deleteContentsRecursive(Util.java:212)
    at hudson.Util.deleteRecursive(Util.java:301)
    at org.jvnet.hudson.test.TestPluginManager$1.run(TestPluginManager.java:146)

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

You can avoid this problem by disabling cache in URLConnection:

- A test class based on HudsonTestCase (junit3 based)


```

public class SomeTest extends HudsonTestCase {
    private boolean origDefaultUseCache = true;

    @Override
    protected void setUp() throws Exception {
        if(Functions.isWindows()) {
            // To avoid JENKINS-4409.
            // URLConnection caches handles to jar files by default,
            // and it prevents delete temporary directories.
            // Disable caching here.
            // Though defaultUseCache is a static field,
            // its setter and getter are provided as instance methods.
            URLConnection aConnection = new File(".").toURI().toURL().openConnection();
            origDefaultUseCache = aConnection.getDefaultUseCaches();
            aConnection.setDefaultUseCaches(false);
        }
        super.setUp();
    }

    @Override
    protected void tearDown() throws Exception {
        super.tearDown();
        if(Functions.isWindows()) {
            URLConnection aConnection = new File(".").toURI().toURL().openConnection();
            aConnection.setDefaultUseCaches(origDefaultUseCache);
        }
    }
}

```

- A test class based on JenkinsRule (junit4 based)

```

public class SomeTest {
    @Rule
    public JenkinsRule j = new JenkinsRule() {
        private boolean origDefaultUseCache = true;

        @Override
        public void before() throws Throwable {
            if(Functions.isWindows()) {
                // To avoid JENKINS-4409.
                // URLConnection caches handles to jar files by default,
                // and it prevents delete temporary directories.
                // Disable caching here.
                // Though defaultUseCache is a static field,
                // its setter and getter are provided as instance methods.
                URLConnection aConnection = new File(".").toURI().toURL().openConnection();
                origDefaultUseCache = aConnection.getDefaultUseCaches();
                aConnection.setDefaultUseCaches(false);
            }
            super.before();
        }

        @Override
        public void after() throws Exception {
            super.after();
            if(Functions.isWindows()) {
                URLConnection aConnection = new File(".").toURI().toURL().openConnection();
                aConnection.setDefaultUseCaches(origDefaultUseCache);
            }
        }
    };
    ...
}

```

%TMP%\hudson-remotingXXXXXXXXXX, %TMP%\jenkins-remotingXXXXXXXXXX, is left even after tests

Those directories contains winp.dll or winp.x64.dll.

That DLL is generated by [winp](#), [Windows process management library](#) and used in `hudson.util.ProcessTree.Windows`.

For the DLL is linked to JVM while classes in winp are loaded, JVM is unable to delete those directories.

This happens only when slaves are launched in tests.

Tests using symbolic links.

- In Windows Vista or later, creating symbolic links needs administrative privilege.
 - You need run tests as an administrator.
- It needs `java.nio.file.Files#isSymbolicLink` to test properly whether a file is a symbolic link in Windows.
 - `java.nio.file.Files` is available in Java ≥ 1.7 .
 - `hudson.Util#isSymlink` handles Java ≥ 1.7 properly. You should use `hudson.Util#isSymlink` rather than `org.apache.commons.io.FileUtils#isSymlink`