Can't build Hadoop 2.4.1 with Java8

The problem is pretty straight-forward. I'm trying to compile Hadoop2.4.1 on windows with the following command :

```
mvn clean package -Pdist,native-win -DskipTests -Dtar
```

With `JAVA_HOME=C:\Program Files\Java\jdk1.7.0_51` , it works fine.

With `JAVA_HOME=C:\Program Files\Java\jdk1.8.0_05` , it doesn't and fails giving me the following error :

```
[INFO] Apache Hadoop Annotations ......................... FAILURE [4.086s]
---
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-javadoc-plugin:2.8
.1:jar (module-javadocs) on project hadoop-annotations: MavenReportException: Er
ror while creating archive:
[ERROR] Exit code: 1 - C:\hadoop-src\hadoop-common-project\hadoop-annotations\sr
c\main\java\org\apache\hadoop\classification\InterfaceStability.java:27: error:
unexpected end tag: </ul>
[ERROR] * </ul>
[ERROR] ^
[ERROR]
[ERROR] Command line was: "C:\Program Files\Java\jdk1.8.0_05\jre\..\bin\javadoc.
exe" -J-Dhttp.proxySet=true -J-Dhttp.proxyHost=proxy -J-Dhttp.proxyPort=3128 @op
tions @packages
[ERROR]
[ERROR] Refer to the generated Javadoc files in 'C:\hadoop-src\hadoop-common-pro
ject\hadoop-annotations\target' dir.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e swit
ch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please rea
d the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionE
xception
[ERROR]
```

As stated before, I changed nothing else than `JAVA_HOME` . The error message seems to indicate that the error is proxy related, but I have no idea why.

In both case, I have

```
C:\hadoop-src>javac -version
javac 1.8.0_05
```

and

```
C:\hadoop-src>java -version
java version "1.8.0_05"
Java(TM) SE Runtime Environment (build 1.8.0_05-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.5-b02, mixed mode)
```

Would you folks have any clues on what's going on ?

maven   hadoop   java-8

edited Jul 7 '14 at 16:43   asked Jul 7 '14 at 16:37

fxm
2,006   21   48

2 Answers

This is an error reported by javadoc. The javadoc version in Java 8 is considerably more strict than the one in earlier version. It now signals an error if it detects what it considers to be invalid markup, including the presence of an end tag where one isn't expected.

To turn off this checking in javadoc, add the `-Xdoclint:none` flag to the javadoc command line. For information about how to do this in a maven environment, see Stephen Colebourne's blog entry on this topic. Specifically, add

```
<additionalparam>-Xdoclint:none</additionalparam>
```

to an appropriate properties or configuration file.

There are a couple weird things going on, though. The current (trunk) version of this file seems to have the `</ul>` end tag in the right place. The history of this file indicates that the previously-missing end tag was added fairly recently, but it does appear to be in Hadoop 2.4. And this file by itself is processed successfully by JDK 8u5 javadoc, without having to suppress any errors.

Has a patch been applied somewhere that added the formerly-missing `</ul>` end tag, which is now redundant since the end tag has been added to the original source? An extra end tag will cause javadoc to fail with this error.

UPDATE

I had been looking at the wrong branch. The 2.4.1 version of this file clearly has an extra `</ul>` end tag. My conjecture about an errant patch was sort-of right. The history of the file on the trunk shows that a bunch of javadoc comments were added back in June 2012 by HADOOP-8059. These new additions were missing the `</ul>` end tag. In January 2014, HADOOP-10320 added the missing end tag. The patch for HADOOP-10320 was transplanted to the 2.4.1 branch, but the new javadocs from HADOOP-8059 were not transplanted, resulting in malformed markup.

edited Jul 8 '14 at 20:27   answered Jul 7 '14 at 22:05

                                        Stuart Marks
                                        44.6k   8   97   136

No patch that I'm aware of. I just downloaded the 2.4.1 source tarball from the apache website (which was not modified since the 21 jun). I checked the same file in the src and there are some other inconsistencies (lack of three imports). The doc is also not the same, and there is indeed a wrongly placed `</ul>` . – fxm Jul 8 '14 at 7:44

@fxm Ah, I've figured it out. Will update the answer. But the `-Xdoclint:none` workaround is the same, unless you want to hack the source file directly. – Stuart Marks Jul 8 '14 at 20:20

Absolutely, I accepted your answer as soon as I managed to compile everything :). Thanks for digging a little more, it's good to know ! – fxm   Jul 9 '14 at 7:05

Alternatively to Stuarts suggestion (I had a difficult time finding out where to put the additionalparam): In order to skip javadoc generation altogether, just run

```
mvn clean package -Pdist,native-win -DskipTests -Dtar -Dmaven.javadoc.skip=true
```

answered Aug 26 '14 at 14:15

                                        Michael Jess
                                        1,397   1   12   15