

Item 121

git_comments:

1. * * Unmarshals * * @param unmarshaller the unmarshaller * @param value the stream to unarmashal (will close it after use, also if exception is thrown) * @return the value * @throws JAXBException is thrown if an exception occur while unmarshalling

git_commits:

1. **summary:** CAMEL-980: Should close stream during unmarshal in case of exception then eg a file stream is open and the file can then not be deleted.
message: CAMEL-980: Should close stream during unmarshal in case of exception then eg a file stream is open and the file can then not be deleted. git-svn-id:
<https://svn.apache.org/repos/asf/activemq/camel/trunk@705344> 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** File component - can not delete consumed file when processing failed (look into real cause)
description: See nabble: <http://www.nabble.com/Problem-with-FileEndpoint-td19940096s22882.html>
Workaround Use `.convertBodyTo(String.class)` in the route so the file is read into memory, and the `java.io.File` handler shouldn't be open/locked when camel attempts to delete it after wards.

jira_issues_comments:

1. I think this is OS dependent. I have tried 1.4.0 and 1.5-SNAPSHOT with and without the `convertBodyTo` and all moves the file from IN.
2. Thanks Christian. Sorry I was a bit grumpy. It was late and I wasn't in the mood to fight with maven.
3. I guess the failure can be simulated by just throwing an exception in the processor instead of all the jaxb failing?
4. **body:** I have attached a complete example for file endpoint with jaxb and bean processor. The `readme.txt` explains how to use the example and how to reproduce this issue. Please write back if the example works and also shows you the issue.
label: documentation
5. Christian when I run your sample using camel 1.4.0 and drop first the ok and then the error file I get:
C:\camelttest>dir /s /b C:\camelttest\in C:\camelttest\ok C:\camelttest\error
C:\camelttest\ok\shiporder_ok.xml C:\camelttest\ok\shiporder_error.xml
C:\camelttest\error\shiporder_error.xml There is no files in the in folder that is locked etc. However you get the error files *twice* and as it's in the ok folder *and* the error folder. We might consider the *moveNamePrefix* option to only work if the processing was a success (no exceptions).
6. Hi Claus, I just updated my checkout and compiled myself. The issue does not occur anymore. Thanks for this fast fix. Greetings Christian
7. Have to add this to pom.xml {code:xml} <pluginRepositories> <pluginRepository> <id>java.net</id>
<name>java.net Maven Repository</name> <url>http://download.java.net/maven/1/</url>
<layout>legacy</layout> <releases> <enabled>true</enabled> </releases> <snapshots>
<enabled>false</enabled> </snapshots> </pluginRepository> </pluginRepositories> {code}
8. I have committed a fix. It should take 2+ hours for the new 1.5-SNAPSHOT to be published to the maven repo. Maybe you can try tomorrow morning.
9. I have attached a new example with a working pom. Sorry .. did not check if it works with an empty repo.
10. **body:** However I am wondering if we should support a more flexible rename strategy so you can - move to a folder if OK - delete if OK - move to another folder if ERROR - delete if ERROR Where you can

mix and match what you need. Well something to consider for Camel 2.0

label: code-design

11. Christian can you either supply a pom.xml that works out-of-the-box and can download all the needed jars. Or switch to another jaxb maven plugin that actually works and is installable. Googling looks like you need to install manually and/or rename all versions to jwsdp. <https://jaxb.dev.java.net/jaxb-maven2-plugin/> <http://maven-plugins.sourceforge.net/maven-jaxb-plugin/> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-api/jwsdp.1.5/jaxb-api-jwsdp.1.5.pom> Downloading: <http://repository.jboss.org/maven2/jwsdp/jaxb-api/jwsdp.1.5/jaxb-api-jwsdp.1.5.pom> Downloading: <http://mirrors.ibiblio.org/pub/mirrors/maven2/jwsdp/jaxb-api/jwsdp.1.5/jaxb-api-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-api/jwsdp.1.5/jaxb-api-jwsdp.1.5.pom> Downloading: <https://maven-repository.dev.java.net/repository/jwsdp/jaxb-api/jwsdp.1.5/jaxb-api-jwsdp.1.5.pom> Downloading: <http://repo1.maven.org/maven2/jwsdp/jaxb-api/jwsdp.1.5/jaxb-api-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-impl/jwsdp.1.5/jaxb-impl-jwsdp.1.5.pom> Downloading: <http://repository.jboss.org/maven2/jwsdp/jaxb-impl/jwsdp.1.5/jaxb-impl-jwsdp.1.5.pom> Downloading: <http://mirrors.ibiblio.org/pub/mirrors/maven2/jwsdp/jaxb-impl/jwsdp.1.5/jaxb-impl-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-impl/jwsdp.1.5/jaxb-impl-jwsdp.1.5.pom> Downloading: <https://maven-repository.dev.java.net/repository/jwsdp/jaxb-impl/jwsdp.1.5/jaxb-impl-jwsdp.1.5.pom> Downloading: <http://repo1.maven.org/maven2/jwsdp/jaxb-impl/jwsdp.1.5/jaxb-impl-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-xjc/jwsdp.1.5/jaxb-xjc-jwsdp.1.5.pom> Downloading: <http://repository.jboss.org/maven2/jwsdp/jaxb-xjc/jwsdp.1.5/jaxb-xjc-jwsdp.1.5.pom> Downloading: <http://mirrors.ibiblio.org/pub/mirrors/maven2/jwsdp/jaxb-xjc/jwsdp.1.5/jaxb-xjc-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-xjc/jwsdp.1.5/jaxb-xjc-jwsdp.1.5.pom> Downloading: <https://maven-repository.dev.java.net/repository/jwsdp/jaxb-xjc/jwsdp.1.5/jaxb-xjc-jwsdp.1.5.pom> Downloading: <http://repo1.maven.org/maven2/jwsdp/jaxb-xjc/jwsdp.1.5/jaxb-xjc-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-lib/jwsdp.1.5/jaxb-lib-jwsdp.1.5.pom> Downloading: <http://repository.jboss.org/maven2/jwsdp/jaxb-lib/jwsdp.1.5/jaxb-lib-jwsdp.1.5.pom> Downloading: <http://mirrors.ibiblio.org/pub/mirrors/maven2/jwsdp/jaxb-lib/jwsdp.1.5/jaxb-lib-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jaxb-lib/jwsdp.1.5/jaxb-lib-jwsdp.1.5.pom> Downloading: <https://maven-repository.dev.java.net/repository/jwsdp/jaxb-lib/jwsdp.1.5/jaxb-lib-jwsdp.1.5.pom> Downloading: <http://repo1.maven.org/maven2/jwsdp/jaxb-lib/jwsdp.1.5/jaxb-lib-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jax-qname/jwsdp.1.5/jax-qname-jwsdp.1.5.pom> Downloading: <http://repository.jboss.org/maven2/jwsdp/jax-qname/jwsdp.1.5/jax-qname-jwsdp.1.5.pom> Downloading: <http://mirrors.ibiblio.org/pub/mirrors/maven2/jwsdp/jax-qname/jwsdp.1.5/jax-qname-jwsdp.1.5.pom> Downloading: <http://download.java.net/maven/2/jwsdp/jax-qname/jwsdp.1.5/jax-qname-jwsdp.1.5.pom>
12. Christian. The jaxb stuff 'com.sun.tools.xjc.maven2' can not be found in the maven repo you have in the pom.xml
13. Sorry forgot about the convertBodyTo(String.class)
14. Christian Well spotted. I am working on a patch to properly close the stream after use. Could you then test it using 1.5-SNAPSHOT on your system to see if it works?
15. Closing 1.5 issues
16. Christian what OS are you using? I am on Windows XP.
17. That is so great. Thanks Christian for taking the time to provide a test app and pin point the issue. I have looked elsewhere and have fixed a few other data formats as well where resources should be closed as well.
18. Christian can you also try 1.5-SNAPSHOT as the file component had some changes as well, also regarding this WARN can not rename. See camel download page for maven snapshot repo <http://activemq.apache.org/camel/download.html>
19. Hi Claus, I am running Windows Vista but I don't think this is a OS dependent issue. Could you reproduce the error when you leave out the convertBodyTo ? I have debugged the error. The problem seems to be in: org.apache.camel.converter.jaxb.FallbackTypeConverter line 90 InputStream inputStream = parentTypeConverter.convertTo(InputStream.class, value); if (inputStream != null) { >>>> Object unmarshalled = unmarshaller.unmarshal(inputStream); return type.cast(unmarshalled); } It opens an inputStream and calls unmarshal. I am not sure how unmarshal should work. Does it close the stream when all goes well? In any case it leaves the stream open if some exception happens. So I think there should be a try {} finally {} in FallbackTypeConverter that tries to close the stream. Of with a try catch on close. This block is probably also necessary for the other variants following. What do you think?

20. **body:** It was indeed quite a fight with maven. There are several different versions out there in several repositories and most do not work. The documentation from the plugin website also does not really help. ;-(About your comment on using a processor to show the failure. I am not sure if it is possible to do so. It depends where the file will be opened. I guess when I simply add a processor after the file endpoint the file will not be opened at all. So I would guess the error will not show.
- label:** documentation