

git_comments:

1. * *
2. * * 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.
3. * * Strategy to create a new {@link FileConsumer} * * @param processor the given processor * @param operations file operations * @return the created consumer

git_commits:

1. **summary:** CAMEL-4004: Introduced newFileConsumer for esier subclassing the file consumer.
message: CAMEL-4004: Introduced newFileConsumer for esier subclassing the file consumer. git-svn-id: <https://svn.apache.org/repos/asf/camel/trunk@1199159> 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Permit to override creation of FileConsumer in FileEndPoint
description: The method FileEndPoint.createConsumer does not permit to override type of FileConsumer. I would use a specific FileConsumer which override prePollCheck method. This will be possible if you externalize method which create FileConsumer in the body of createConsumer :
{code}FileConsumer result = new FileConsumer(this, processor, operations);{code} could be refactor in :
{code}FileConsumer result = newFileConsumer(processor, operations); ... protected FileConsumer newFileConsumer(Processor processor, GenericFileOperations<File> operations){ return new FileConsumer(this, processor, operations) }{code}

jira_issues_comments:

1. **body:** Can you provide a patch and an unit test?
label: test
2. Anyone in our community want to provide a patch for this? Mind that there is also the camel-ftp component which may need to be adjusted.
3. Claus, you simply take over all those tricky & tough stuff for yourself to fix (namespace aware xml streaming tokenizier support, just to mention one of those last features), and leave us alone with such trivial fixes... this's simply not fair ;-)) I see a good match here of your definition for the professionalism and the action you took today by this ticket: <http://camel.465427.n5.nabble.com/HEADS-UP-Adjustments-to-ExecutorServiceManager-on-trunk-tp4693698p4715407.html> Regards, Babak
4. **body:** Sorry for the delay, but I can't provide a unit test, but I've given to you a patch in the description of this JIRA. The solution is to externalize the FileConsumer creation into a new method which could be overloaded. Then you could use a different implementation of FileConsumer, like below: {code} public class PollConditionFileEndpoint extends FileEndpoint { @Override protected FileConsumer newFileConsumer(Processor processor, FileOperations operations) { return new FileConsumer(this, processor, operations){ @Override protected boolean prePollCheck() throws Exception { return

`Boolean.parseBoolean("customizable condition"); } } } } {code}` So, could you re-open this JIRA ?

Thanks

label: test

5. **body:** Well maybe instead of a `newFileConsumer` it would be better to offer an API for easier providing custom `prePoll` / `postPoll` logic to the file consumer? Then people do not have to extend components in Camel. That said I went ahead and created the `newFileConsumer` + unit test. We do love contributions. Read here. Also how to provide patches, as ASF have policy for how thats to happen: <http://camel.apache.org/contributing.html> If you are eager to get this into the code base, then helping out yourself is great for the community. And after the first patch, it becomes easier to submit the next patch ...

label: code-design

6. I also need to create different consumers based on the options provided in the new component I'm working on. Any proposals how the API could be extended for that purpose in a generic way applicable for all component?
7. For all components? Not just a file based component? If so then I suggest to use `@dev` to talk about such, as its not related very much to this JIRA ticket.