

#### git\_comments:

1. \* \* A simple configuration wrapper for Apache Commons configuration. The implementation supports \* reading and writing properties style config and interpolation. \* \* <p> \* This simple wrapper was created to keep 3rd party APIs out of the Fluo API. \* \* @since 1.0.0
2. \* \* Read a properties style config from given input stream.
3. \* \* @return raw property without interpolation or null if not set.
4. **comment:** \* \* Read a properties style config from given file.  
**label:** code-design
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. \* \* Returns a subset of config that start with given prefix. The prefix will not be present in keys \* of the returned config. Any changes made to the returned config will be made to this and visa \* versa.
7. \* \* Copy constructor.

#### git\_commits:

1. **summary:** fixes #695 removed commons config from public API  
**message:** fixes #695 removed commons config from public API  
**label:** code-design

#### github\_issues:

1. **title:** Remove third party types from public API  
**body:** I have been experimenting with classloaders to separate the Fluo frameworks dependencies from application dependencies which executing Observers. One thing I have realized as part of this experimenting is that having commons config in the API makes this dependency isolation more difficult. It forces the Fluo and a Fluo user to have the same commons config. In order to support the possibility of dependency isolation in the future I think it would be best if FluoConfiguration encapsulated commons config instead of extending it. Also need to check if public API uses any other 3rd party types.  
**label:** code-design
2. **title:** Remove third party types from public API  
**body:** I have been experimenting with classloaders to separate the Fluo frameworks dependencies from application dependencies which executing Observers. One thing I have realized as part of this experimenting is that having commons config in the API makes this dependency isolation more difficult. It forces the Fluo and a Fluo user to have the same commons config. In order to support the possibility of dependency isolation in the future I think it would be best if FluoConfiguration encapsulated commons config instead of extending it. Also need to check if public API uses any other 3rd party types.

#### github\_issues\_comments:

#### github\_pulls:

#### github\_pulls\_comments:

#### github\_pulls\_reviews:

#### jira\_issues:

#### jira\_issues\_comments: