# Use Apache Commons IO for directory monitoring

## Context and Problem Statement

In JabRef, there is a need to add a directory monitor that will listen for changes in a specified directory.

Currently, the monitor is used to automatically update the LaTeX Citations when a LaTeX file in the LaTeX directory is created, removed, or modified (#10585). Additionally, this monitor will be used to create a dynamic group that mirrors the file system structure (#10930).

## Considered Options

- Use java.nio.file.WatchService
- Use io.methvin.watcher.DirectoryWatcher
- Use org.apache.commons.io.monitor

## Decision Outcome

Chosen option: “Use org.apache.commons.io.monitor”, because comes out best (see below).

## Pros and Cons of the Options

### java.nio.file.WatchService

- Good, because it is a standard Java API for watching directories.
- Good, because it does not need polling, it is event-based for most operating systems.
- Bad, because:
  1. Does not detect files coming together with a new folder (JDK issue: JDK-8162948).
  2. Deleting a subdirectory does not detect deleted files in that directory.
  3. Access denied when trying to delete the recursively watched directory on Windows (JDK issue: JDK-6972833).
  4. Implemented on macOS by the generic `PollingWatchService`. (JDK issue: JDK-8293067)

### io.methvin.watcher.DirectoryWatcher

- Good, because it implemented on top of the `java.nio.file.WatchService`, which is a standard Java API for watching directories.
- Good, because it resolves some of the issues of the `java.nio.file.WatchService`.
    - Uses`ExtendedWatchEventModifier.FILE_TREE` on Windows, which resolves issues (1, 3) of the `java.nio.file.WatchService`.
    - On macOS have native implementation based on the Carbon File System Events API, this resolves issue (4) of the `java.nio.file.WatchService`.
- Bad, because issue (2) of the `java.nio.file.WatchService` is not resolved.

### org.apache.commons.io.monitor

- Good, because there are no observed issues.
- Good, because can handle huge amount of files without overflowing.
- Bad, because it uses a polling mechanism at fixed intervals, which can waste CPU cycles if no change occurs.

---