

//

commons-ioFileEntrycommons-io

aFileResourceIOUtilsResourceRepository

bResourceEntry

Entry

```
public boolean refresh(final String path) {  
  
    //  
    final boolean origExists = exists;  
    final long origLastModified = lastModified;  
    final boolean origDirectory = directory;  
    final long origLength = length;  
  
    //  
    name = ResourceIOUtils.getName(path);  
    exists = ResourceIOUtils.exist(path);  
    directory = exists && ResourceIOUtils.isDirectory(path);  
    lastModified = exists ? ResourceIOUtils.lastModified(path) : 0;  
    length = exists && !directory ? ResourceIOUtils.getLength(path) : 0;  
  
    //  
    return exists != origExists ||  
           lastModified != origLastModified ||  
           directory != origDirectory ||  
           length != origLength;  
}
```

refreshrefresh

3

```

//
ResourceAlterationObserver observer = new ResourceAlterationObserver("reportlets");
//
observer.addListener(new BaseResourceAlterationListener() {
    @Override
    public void onStart(ResourceAlterationObserver observer) {
        System.out.println("start...");
    }

    @Override
    public void onDirectoryCreate(String directory) {
        System.out.println("DirectoryCreate: " + directory);
    }

    @Override
    public void onDirectoryChange(String directory) {
        System.out.println("DirectoryChange: " + directory);
    }

    @Override
    public void onDirectoryDelete(String directory) {
        System.out.println("DirectoryDelete: " + directory);
    }

    @Override
    public void onFileCreate(String file) {
        System.out.println("FileCreate: " + file);
    }

    @Override
    public void onFileChange(String file) {
        System.out.println("FileChange: " + file);
    }

    @Override
    public void onFileDelete(String file) {
        System.out.println("FileDelete: " + file);
    }

    @Override
    public void onStop(ResourceAlterationObserver observer) {
        System.out.println("stop");
    }
});
//
//1000ms
ResourceAlterationMonitor monitor = new ResourceAlterationMonitor(1000, observer);
//-1
//ResourceAlterationMonitor monitor = new ResourceAlterationMonitor(-1, observer);

//
monitor.addObserver(observer);

try {
    //
    //interval == -1
    //monitor.runOnce();
    //interval > 0 start
    monitor.start();
} catch (Exception e) {
    e.printStackTrace();
}

```