

The Java™ Tutorials

Trail: Essential Classes

Lesson: Basic I/O

Section: File I/O (Featuring NIO.2)

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases.

Moving a File or Directory

You can move a file or directory by using the `move(Path, Path, CopyOption...)` method. The move fails if the target file exists, unless the `REPLACE_EXISTING` option is specified.

Empty directories can be moved. If the directory is not empty, the move is allowed when the directory can be moved without moving the contents of that directory. On UNIX systems, moving a directory within the same partition generally consists of renaming the directory. In that situation, this method works even when the directory contains files.

This method takes a varargs argument – the following `StandardCopyOption` enums are supported:

- `REPLACE_EXISTING` – Performs the move even when the target file already exists. If the target is a symbolic link, the symbolic link is replaced but what it points to is not affected.
- `ATOMIC_MOVE` – Performs the move as an atomic file operation. If the file system does not support an atomic move, an exception is thrown. With an `ATOMIC_MOVE` you can move a file into a directory and be guaranteed that any process watching the directory accesses a complete file.

The following shows how to use the `move` method:

```
import static java.nio.file.StandardCopyOption.*;
...
Files.move(source, target, REPLACE_EXISTING);
```

Though you can implement the `move` method on a single directory as shown, the method is most often used with the file tree recursion mechanism. For more information, see [Walking the File Tree](#).