

The Java™ Tutorials

Trail: Essential Classes

Lesson: Basic I/O

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.

File I/O (Featuring NIO.2)

Note: This tutorial reflects the file I/O mechanism introduced in the JDK 7 release. The Java SE 6 version of the File I/O tutorial was brief, but you can download the [Java SE Tutorial 2008-03-14](#) version of the tutorial which contains the earlier File I/O content.

The `java.nio.file` package and its related package, `java.nio.file.attribute`, provide comprehensive support for file I/O and for accessing the default file system. Though the API has many classes, you need to focus on only a few entry points. You will see that this API is very intuitive and easy to use.

The tutorial starts by asking [what is a path?](#) Then, the [Path class](#), the primary entry point for the package, is introduced. Methods in the `Path` class relating to [syntactic operations](#) are explained. The tutorial then moves on to the other primary class in the package, the `Files` class, which contains methods that deal with file operations. First, some concepts common to many [file operations](#) are introduced. The tutorial then covers methods for [checking](#), [deleting](#), [copying](#), and [moving](#) files.

The tutorial shows how [metadata](#) is managed, before moving on to [file I/O](#) and [directory I/O](#). [Random access files](#) are explained and issues specific to [symbolic and hard links](#) are examined.

Next, some of the very powerful, but more advanced, topics are covered. First, the capability to [recursively walk the file tree](#) is demonstrated, followed by information about how to [search for files using wild cards](#). Next, how to [watch a directory for changes](#) is explained and demonstrated. Then, [methods that didn't fit elsewhere](#) are given some attention.

Finally, if you have file I/O code written prior to the Java SE 7 release, there is a [map from the old API to the new API](#), as well as important information about the `File.toPath` method for developers who would like to [leverage the new API without rewriting existing code](#).

[About Oracle](#) | [Contact Us](#) | [Legal Notices](#) | [Terms of Use](#) | [Your Privacy Rights](#)

Copyright © 1995, 2017 Oracle and/or its affiliates. All rights reserved.

Previous page: Object Streams

Next page: What Is a Path? (And Other File System Facts)