

Files

C4

Java File IO

Streams

Standard IO

Random access files

Buffering

File IO as Streams

A **stream** is a standard abstraction used for files:

- A sequence of values are read.

- A sequence of values are written.

The stream reflects the sequential nature of file IO and the physical characteristics of the media on which files traditionally reside (e.g. tape or a spinning disk).





Java I/O: Byte Streams

The classes `InputStream` and `OutputStream` allow you to read and write streams of bytes to and from streams including files (subclasses: `FileInputStream` and `FileOutputStream`).

- Open the stream
 - Read or write from the stream (in **bytes**)
 - Wrap operations in a **try** clause
 - Use **finally** to close the streams
- ints** are used, even though **bytes** are transferred(!)

Java I/O: Character Streams

When reading and writing characters, you should use the classes `Reader` and `Writer`, which allow you to read and write streams of characters to and from streams including files (subclasses: `FileReader` and `FileWriter`).

`ints` are used, even though `chars` are transferred.

File I/O: Buffering

Reading data one byte at a time is costly. Buffering is used to absorb some of that overhead.

Disk: ~10ms SSD: ~100 μ s RAM: ~10ns Register: ~1ns

In Java the `BufferedReader` and `BufferedWriter` classes can be used to buffer data read or written with `FileReader` and `FileWriter`.

To be sure that a buffer is flushed, call `flush()`, or close the file.

Java Command Line IO

Three standard IO streams (globally-defined objects):

- Standard input `System.in`
- Standard output `System.out`
- Standard error `System.err`

```
byte b = (byte) System.in.read();  
System.out.write(b);  
System.out.flush();  
System.err.write(b);
```


“New” I/O (`java.nio.file`)

Java NIO offers simpler, event-driven interface

- `Path` — replaces `java.io.File`
- `FileSystem` — factory class for objects in the filesystem
- `WatchService` — utility class to detect file system changes through event notification
- `Files` — create, rename, copy, modify attributes and delete files