

HowToDoInJava

[Interview Questions](#)[Popular Guides](#)[Core Java](#)[More Topics](#)

Java NIO – Memory-Mapped Files with

Memory-Mapped Files MappedByteBuffer

If you know [how java IO works at lower level](#), then you will be aware of buffer handling, memory paging and other such concepts. For conventional file I/O, in which user processes issue `read()` and `write()` system calls to transfer data, there is almost always one or more copy operations to move the data between these filesystem pages in kernel space and a memory area in user space. This is because there is not usually a one-to-one alignment between filesystem pages

and the type of I/O operation supported by most hardware. To take maximum advantage of the hardware, we usually avoid buffer copies. This is called **memory-mapped files**. Things here around memory-mapped files.

To establish a virtual memory mapping from user space to kernel space, we use a memory-mapped file. With a memory-mapped file, you can access it by simply treating it as a regular file. This simplifies the code you write in order to modify the



Be Awesome When It Matters !!

Join 4000+ Happy Learners and receive my free ebook "**5 Steps Guide for Java Interview Preparation**". This book will help you in taking your interview preparation to next stage, where you will **outshine your competitors**.

Sign up to download !!

SUBSCRIBE

napped files, we start with a **RandomAccessFile**, byte buffers are created via the **ByteBuffer** class with operations that are mapped byte buffer and the file mapping that it is garbage-collected. Note that you must specify the region that you want to map in the file; this is for smaller regions of a large file.

```

128 * 1024 * 1024 / 1024 = 128 Mb
2) throws Exception

RandomAccessFile("howtodoinjava.dat", "rw")
FileChannel().map(FileChannel.MapMode.READ_WRITE, 0,

128 * 1024 * 1024);

```



128 MB long, which is probably larger than the memory available to be accessible all at once because only portions of the file are swapped out. This way a very large file (up to 2 GB) can be mapped.

FileChannel.MapMode can be writable or read-only. The first two modes, **MapMode.READ_WRITE** and **MapMode.READ_ONLY**, are fairly obvious. **MapMode.READ_ONLY** is used to be read-only or to allow modification of the mapped buffer. **MapMode.PRIVATE**, indicates that you want a copy-on-write mapping. Any changes you make via `put()` will result in a private copy of the buffer that your instance can see. No changes will be made to the underlying file. If you want to make any changes to the underlying file, you must use a **MapMode.PRIVATE** mapping. This is done by creating a **ByteBuffer** object to allow `put()`s.

Once established, a mapping remains in effect until the **ByteBuffer** is garbage collected.

ge collected. Also, mapped buffers are not tied to associated FileChannel does not destroy the itself breaks the mapping.

size, but the file it's mapped to is elastic. While the mapping is in effect, some or all of the defined data could be returned, or unchecked about how files are manipulated by other threads memory-mapped.

-Mapped Files

es over normal I/O:

s memory, so there is no need to issue `read()` or

mapped memory space, page faults will be generated from disk. If the user modifies the mapped automatically marked as dirty and will be te the file.

operating system will perform intelligent caching g memory according to system load.

no buffer copying is ever needed.

ut consuming large amounts of memory to copy

oped File

e below code template:

```
"bigFile.xls";
) throws Exception
);
```

```

ide
domAccessFile(file, "r").getChannel();

using channel.map() operation
nnel.map(FileChannel.MapMode.READ_ONLY, 0, file

as if it were loaded in memory.
d()); //prints false
y()); //Get the size based on content size of

buffer the way you like.
); i++)

r.get()); //Print the content of file

```



Mapped File

ped IO, use below code template:

```

{
"test.txt";

) throws Exception {

);

a new file

ode
domAccessFile(file, "rw").getChannel();

using channel.map() operation
nnel.map(FileChannel.MapMode.READ_WRITE, 0, 4096

hods
tBytes());

```



omments section.

vs for FREE !!!

Get my free ebook "**5 Steps Guide for Java**
to help you in taking your interview preparation
your competitors.

SUBSCRIBE

a

started howtodojava.com in late 2012. I love computers,
solving problems everyday. A family guy with fun
things to share. You can find me on [my ebook](#), [Twitter](#) and [Google Plus](#).

: 2GB. Can I define the offset and split the file
successful doing it. Can you please throw your

42 pm

ng a Memory-Mapped File”? How bog file it was

appedBus
, which supports having multiple processes
y mapped file as well as having multiple

a lot.

bytes in large file and I was using a
methods and it seems that the MappedByteBuffer

Feedback

required fields are marked *

java ... **/java** tags otherwise code may not



Search Tutorials

Java Input/Output

[Java IO Tutorial](#)[How I/O Works Internally?](#)[Standard IO vs. NIO](#)[Automatic Resource Mgmt](#)[Copy Directory](#)[Delete Directory with Files](#)[Create File](#)[Read File – BufferedReader](#)[Read File – Byte\[\]](#)[Read File – LineNumberReader](#)[Read File – Java 8 Streams](#)[Read File – NIO](#)[Read/Write – CSV Files](#)[Read/Write – Properties Files](#)[Read/Write – Excel Files](#)[Read Only File](#)[Check if File Exist](#)[Copy File – 4 Methods](#)[Delete File – FilenameFilter](#)[FileFilter Example](#)[Create Temporary File](#)[Write to Temporary File](#)[Delete Temporary File](#)[Read Input from Console](#)[Convert String to InputStream](#)[Convert InputStream to String](#)

[Password Protected Zip Files](#)

[Manage System Log Size](#)

[Generate SHA or MD5 Checksum](#)

[Deep Cloning using Serialization](#)

[Base64 Encode/Decode – Java 8](#)

Java NIO

[Java – NIO Introduction](#)

[Java NIO vs. Standard IO](#)

[Java NIO – Path](#)

[Java NIO – Buffers](#)

[Java NIO – Channels](#)

[Java NIO – Read Files](#)

[Java NIO – Streams of Lines](#)

[Java NIO – MappedByteBuffer](#)

[Java NIO – Vectored IO](#)

[Java NIO – Data Transfer between Channels](#)

Recommended

[10 Life Lessons](#)

[Generate Secure Passwords](#)

[How Web Servers work?](#)

[How Java I/O Works Internally?](#)

[5 Class Design Principles](#)

[Unit Testing Best Practices](#)

[Exception Handling Best Practices](#)

[Best Way to Learn Java](#)

[Java Best Practices](#)

[Java Interview Questions](#)

[Best Java Books](#)

Developer Tools

[JSON Formatter and Minifier](#)

[XML Formatter and Minifier](#)

[CSS Formatter and Minifier](#)

[HTML Formatter and Minifier](#)

Be Informed And Stay Updated !!

Join **4000+ Happy Learners** and receive my free ebook **"5 Steps Guide for Java Interview Preparation"**. This book will help you in taking your interview preparation to next stage, where you will **outshine your competitors**.

Sign up to download !!

SUBSCRIBE

References

[Oracle Java Docs](#)
[Spring 3 Reference](#)
[Spring 4 References](#)
[Jboss RESTEasy JAX-RS](#)
[Hibernate Developer Guide](#)
[JUnit Wiki](#)
[Maven FAQs](#)
[Dzone Dev Links](#)
[JSP Homepage](#)
[ANT Homepage](#)

Meta Links

[Advertise](#)
[Contact Us](#)
[Privacy policy](#)
[About Me](#)
[Online tests](#)

Copyright © 2016 · HowToDoInJava.com · All Rights Reserved